Thèse d'habilitation à diriger des recherches "Analysis of Comparison-based Stochastic Continuous Black-Box Optimization Algorithms"

Anne Auger

Université Paris-Sud

Faculté des sciences d'Orsay

École doctorale de mathématiques de la région Paris-Sud (ED 142)

Mémoire présenté pour l'obtention du

Diplôme d'habilitation à diriger des recherches

Discipline : Mathématiques

par

Anne AUGER

Analysis of Comparison-based Stochastic Continuous Black-Box Optimization Algorithms

---

Date de soutenance : 5 Février 2016

Composition du jury :

Stéphane Gaubert, Inria / CMAP, École Polytechnique	(Examinateur)
Christian Igel, University of Copenhagen	(Rapporteur)
Eric Moulines, CMAP, École Polytechnique	(Rapporteur)
Yann Ollivier, CNRS / LRI, Université Paris-Sud	(Rapporteur)
Olivier Pironneau, LJLL, Université Pierre et Marie Curie	(Examinateur)
Filippo Santambrogio, LMO, Université Paris-Sud	(Examinateur)
Marc Schoenauer, Inria / LRI, Université Paris-Sud	(Examinateur)

Dedicated to my grandmother

Chapter 1

Acknowledgments

I wish to warmly thank Christian Igel, Eric Moulines and Yann Ollivier for kindly taking the time to review my habilitation manuscript and Stéphane Gaubert, Olivier Pironneau, Filippo Santambrogio for kindly agreeing to be part of the habilitation jury.

There are a few persons that have strongly influenced the (not purely) random path I have followed to arrive at the happy place where I am today that I wish to thank.

First of all, Marc Schoenauer has been supporting me since my beginning in research. I am particularly deeply grateful for encouraging me to pursue in academia and supporting me for obtaining an Inria position.

Second, Nikolaus Hansen is a remarkable scientist with whom I had the chance to collaborate and learn from for many years. I feel that some of our joint work (modestly) contribute in improving the world. This is one of the deep down reason I wake up every day feeling happy to go to work ...

Last, on a personal side, I thank Dimo, Flora and Bastian who are filling my life with much joy and happiness.

Antony, February 2017

Contents

Chapter 2

Introduction en français

Ce mémoire décrit l'essentiel de mon travail scientifique depuis la fin de ma thèse. Mes travaux sont centrés sur l'optimisation numérique dite "boîte-noire" à l'exception d'un article effectué durant mon séjour post-doctoral à l'ETH Zurich qui introduit un nouvel algorithme d'optimisation stochastique pour simuler des systèmes en chimie ou bio-chimie [23].

Les algorithmes d'optimisation au coeur de mon travail sont des algorithmes adaptatifs sans-dérivées et stochastiques. Ils sont particulièrement adaptés à l'optimisation de problèmes difficiles dans des contextes où la fonction n'est accessible qu'à travers une "boîte-noire" retournant l'information d'ordre zero, c'est-à-dire que la seule information disponible et utilisable par l'algorithme sont les couples (points de l'espace de recherche, valeur de fonction objectif associée). Ce contexte est très courant dans l'industrie où les problèmes d'optimisation rencontrés font appel à des codes de simulations numériques pour lesquels, souvent, simplement un executable du code est disponible. L'aspect "sans-dérivées" est aussi très commun car le calcul d'un gradient (qui présuppose la fonction sous-jacente dérivable) sur des codes de simulations numériques, par exemple en utilisant une méthode d'adjoint ou de differentiation automatique peut être couteux en temps de développement. Il est par ailleurs usuel que la formulation d'un problème d'optimisation change au fur et à mesure de sa résolution, adapter le code de calcul de gradient peut alors s'avérer très lourd et peut motiver l'utilisation d'une méthode d'optimisation boîte-noire.

Ce contexte d'optimisation boîte-noire s'appelle également optimisation sans dérivées dans la communauté "mathematical programming" et l'acronyme anglais associé est DFO pour "derivative free optimization". Les méthodes qualifiées de DFO sont généralement *deterministes*. Les méthodes DFO les plus connues à l'heure actuelle sont l'algorithme du simplexe ou de Nelder-Mead [79, 77], les algorithmes de "pattern search" [54, 90, 6] et l'algorithme NEWUOA (NEW Unconstraint Optimization Algorithm) développé par Powell [82, 81]. Ce dernier algorithme est à l'heure actuelle considéré comme l'algorithme DFO déterministe état de l'art.

Mon travail porte ainsi sur des méthodes DFO au sens littéral du terme. En revanche, les méthodes auxquelles je me suis intéressées ont une large composante stochastique et ont été développées dans la communauté des algorithmes bio-inspirés qui se compose essentiellement d'ingénieurs et d'informaticiens. Les premiers algorithmes ont été introduits dans les années 70. Un parallèle entre la théorie de Darwin de l'évolution des espèces et l'optimisation a servi à l'origine de source d'inspiration pour leur développement. A l'heure actuelle, ce domaine des méthodes bio-inspirées est également appelé "Evolutionary Computation". Un terme générique pour les algorithmes est algorithme évolutionnaire (EA). Pour beaucoup de chercheurs (dont je fais partie) dans ce domaine, l'aspect bio-inspiré n'est plus présent et le développement des algorithmes est seulement motivé par des considérations mathématiques et numériques.

Parmi les algorithmes évolutionnaires, les algorithmes génétiques (GA) sont probablement encore les plus célèbres en dehors de la communauté EC. En revanche, les GAs ne sont pas des algorithmes compétitifs pour l'optimisation numérique–ce fait est reconnu depuis plus d'une dizaine d'années. Les strategies d'évolutions (ES), introduites à la fin des années 70 [83], se sont

imposées comme les algorithmes évolutionnaires pour l'optimisation numérique. A l'heure actuelle, l'algorithme ES le plus abouti est l'algorithme Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [50]. L'algorithme adapte un vecteur Gaussien (paramétré par vecteur moyenne et matrice de covariance) qui encode la métrique sous-jacente. Cette métrique apprend sur des fonctions convexes quadratiques l'information d'ordre 2, c'est à dire que la matrice de covariance devient proportionnelle à l'inverse de la matrice Hessienne. Ainsi, CMA-ES peut être vu comme le pendant stochastique d'une méthode de quasi-Newton. Une particularité essentielle de CMA-ES et des ES en général est dû au fait qu'ils n'utilisent que des comparaisons pour les différentes mises à jour. Plus précisément, nous avons vu que les ESs sont des algorithmes d'optimisation sans dérivées, ils n'utilisent cependant qu'une information "dégradée" de ce que la boîte-noire leur fournit, à savoir simplement le résultat de la comparaison des solutions candidates, i.e. étant donné deux solutions $\mathbf{x}_1$ et $\mathbf{x}_2$, est ce que $f(\mathbf{x}_1)$ est plus grand ou plus petit que $f(\mathbf{x}_2)$. En conséquence ils optimisent de la même façon une fonction $f : \mathbb{R}^n \to \mathbb{R}$ ou n'importe quelle fonction $g \circ f$ où $g : f(\mathbb{R}^n) \to \mathbb{R}$ est une fonction strictement croissante: ils sont *invariants* à la composition à gauche par une fonction monotone strictement croissante.

L'algorithme CMA-ES est reconnu comme la méthode état de l'art pour l'optimisation stochastique numérique. Il est utilisé dans de nombreuses applications dans l'industrie ou dans le monde académique.

Pour des raisons historiques, les algorithmes ESs ont été développés dans la communauté EC où la mise au point d'un algorithme est la plupart du temps découplée du soucis de prouver un théorème de convergence sur la méthode et repose essentiellement sur l'utilisation de modèles mathématiques approximatifs simplifiés et de simulations numériques sur des fonctions tests. Bien que ce découplage entre mise au point pratique et théorie puisse être vu comme un inconvenient, il présente l'avantage que le développement d'une méthode n'est pas restreinte (ou bridée) par une contrainte technique liée à une preuve mathématique. Cela a permis à un algorithme comme CMA-ES de voir le jour bien avant que l'on comprenne certains de ses fondements théoriques et bien avant que l'on puisse établir une preuve de convergence. En revanche, cela implique aussi que les études théoriques de convergence par exemple s'avèrent relativement compliquées.

Ma recherche se situe dans ce contexte général: je suis particulièrement intéressée par l'étude mathématique d'algorithmes adaptatifs stochastiques comme les algorithmes ESs (en particulier CMA-ES) et par l'établissement de preuves de convergence. Ces algorithmes ont une particularité attractive: bien qu'introduits dans un contexte où les performances pratiques sont plus importantes que les preuves théoriques, ils s'avèrent avoir des fondements mathématiques profonds liés en particulier aux notions d'invariance et de géométrie de l'information. Par ailleurs, ils s'inscrivent dans le cadre plus général d'algorithmes d'approximation stochastique et ils sont fortement connectés aux méthodes Monte-Carlo par chaînes de Markov (MCMC). Ces deux derniers points fournissent des outils mathématiques puissants pour établir des preuves de convergence (linéaire). La comprehension de ces fondements et connexions est reliée en partie à mon travail comme cela sera illustré dans ce mémoire.

J'ai abordé plusieurs facettes de l'optimisation numérique. Bien que l'essentiel de mes travaux porte sur l'optimisation mono-objectif, i.e. minimizer $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$, j'ai également travaillé en optimisation multi-objectif, i.e. où l'on s'intéresse à minimiser une fonction vectorielle $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}^k$. Dans ce cas là, la notion d'optimum est remplacée par celle d'ensemble de points de Pareto composé des meilleurs compromis possibles. Mes contributions portent sur l'étude d'algorithmes à base d'hypervolume qui quantifient la qualité d'un ensemble de solutions en calculant le volume compris entre les solutions et un point de reference. Les algorithmes utilisant l'hypervolume sont à l'heure actuelle les algorithmes état de l'art. Nous avons pu établir des caractérisations théoriques de l'ensemble des solutions optimales au sens de l'hypervolume. En optimisation mono-objectif, j'ai travaillé sur l'optimisation bruitée où étant donné un point de l'espace de recherche, on observe une distribution de valeurs de fonction objectif, sur l'optimisation à grande échelle où l'on s'intéresse à l'optimisation de problèmes avec de l'ordre de 10^4 à 10^6 variables et sur l'optimisation sous contrainte.

Mes travaux s'articulent autour de trois grands axes: théorie / nouveaux algorithmes / appli-

cations (voir Figure 3.1). Ces trois axes sont complémentaires et couplés: par exemple, la mise au point de nouveaux algorithmes repose sur l'établissement de bornes théoriques de convergence et est ensuite complémentée par des simulations numériques. Ceci est illustré au Chapitre 6. Par ailleurs le développement d'algorithmes pour l'optimisation en grande dimension repose sur la connexion entre CMA-ES et la géométrie de l'information (voir Chapitre 4). Un autre exemple de complémentarité est le suivant: les applications abordées notamment pour l'optimisation du placement de puits de pétrole ont motivé l'introduction de nouvelles variantes de CMA-ES (voir Chapitre 9).

Par ailleurs, une partie non négligeable de mes travaux porte sur le test (benchmarking) d'algorithmes. La motivation principale est d'améliorer les méthodologies pour tester et comparer les algorithmes d'optimisation numériques. Ces travaux ont été accompagnés du développement d'une plateforme, Comparing COntinuous Optimizers (COCO) et ont un impact maintenant sur la mise au point de nouveaux algorithmes mais également sur le test d'hypothèses théoriques.

2.1 Organisation

Ce mémoire est organisé autour de six chapitres principaux (en plus des chapitres d'introduction) qui présentent l'essentiel de mes travaux de recherches depuis la fin de ma thèse. L'accent est mis sur les aspects les plus théoriques portants sur les algorithmes stochastiques continus à base de comparaison. Je présente par ailleurs une introduction générale avancée à l'optimisation numérique boîte-noire. Cette introduction est certainement biaisée et reflète les méthodes et concepts que je trouve les plus importants. Elle utilise le materiel de certains papiers dont je suis (co)-auteur.

De manière plus précise, le chapitre 4 est une introduction à l'optimisation numérique boîte-noire sans-dérivées présentant en particulier l'algorithme CMA-ES et les connexions entre algorithmes adaptatifs d'optimisation stochastiques et la géométrie de l'information. Les définitions introduites dans ce chapitre sont utilisées dans les autres chapitres du mémoire. Le chapitre 5 porte sur la notion d'invariance en optimisation et plus particulièrement sur les invariants associés aux méthodes d'optimisation à base de comparaison. C'est également un chapitre général complétant le chapitre précédent. Le chapitre présente également une preuve de l'invariance par transformation affine de l'espace de recherche de l'algorithmes CMA-ES. Le chapitre 6 présente des bornes de convergence pour certains algorithmes ES. Nous illustrons ensuite comment ces bornes servent pour la mise au point de nouveaux algorithmes. Au chapitre 7 nous présentons une méthodologie générale pour prouver la convergence linéaire d'algorithmes adaptatifs à base de comparaison, restreinte au cas de l'adaptation d'un point de l'espace de recherche et d'un pas. Cette méthodologie exploite certains invariants des algorithmes, à savoir invariant par changement d'échelle et invariance par translation, et exhibe une chaîne de Markov sous-jacente candidate à être stable. L'étude de la stabilité de cette chaîne de Markov (irreductibilité, récurrence au sens de Harris et positivité) permet ensuite de conclure à la convergence linéaire de l'algorithme d'optimisation sous-jacent. Cette méthodologie est appliquée pour prouver la convergence linéaire d'un des plus vieux algorithmes d'optimisation stochastique. Ce chapitre présente également la connexion entre algorithmes adaptatifs à base de comparaison pour l'optimisation boîte-noire et algorithmes MCMC. Au chapitre 8, nous présentons d'autres résultats qui exploitent également la théorie des chaînes de Markov à temps discret et sur un espace d'état continu pour analyser certains algorithmes ESs dans un contexte bruité où d'optimisation sous contrainte. Au chapitre 9, nous présentons des contributions moins théoriques ou non reliées à l'optimisation mono-objectif portant sur l'optimisation multi-objectif, le test d'algorithme et l'optimisation du placement de puits de pétroles.

Note: Les références des papiers dont je suis co-auteur apparaissent en cyan, par exemple [22] réfère à un papier dont je suis auteur alors que [79] réfère à un papier dont je ne suis pas auteur.

2.2 Travaux reliés aux thèses encadrées

La plupart des travaux présentés dans ce mémoire ont été effectués en collaboration. Plusieurs résultats sont reliés à des thèses ou travaux de postdoctorats que j'ai co-encadrés (voir Figure 3.1). Je décris brièvement ci-dessous le contenu des thèses co-encadrées et l'endroit où ces travaux sont décrits dans ce mémoire.

Thèse de Mohamed Jebalia (2004 – 2008) encadrée à partir de Octobre 2006, "Optimization using Evolution Strategies : Convergence and convergence rates for noisy functions - Resolution of an Identification Problems". Les études théoriques dans cette thèse portent sur des algorithmes ES et supposent un modèle adaptatif de pas optimal, proportionnel à la distance à l'optimum. Dans l'article [61], nous avons prouvé que des bornes de convergence pour les ES sont reliées à cet algorithme à pas optimal. Ces résultats sont présentés au chapitre 6. La convergence linéaire de l'algorithme à pas optimal a été ensuite étudiée dans le cas d'une fonction sphérique avec bruit multiplicatif [60]. Ces résultats sont présentés au chapitre 8.

Thèse de Zyed Bouzarkouna (Dec. 2008 – Avril 2012), "Well placement optimization": thèse en collaboration avec l'institut Français du Pétrole (IFP) portant sur l'optimisation du placement de puits de pétrole. Dans ce contexte plusieurs algorithmes couplant CMA-ES et meta-modèles ont été proposés [32, 29, 30, 31]. Ces travaux sont présentés rapidement au chapitre 9.

Thèse d'Alexandre Chotard (2011-2015): thèse qui porte sur l'étude théorique d'algorithmes ES et en particulier sur plusieurs analyses de convergence à l'aide de la théorie des chaînes de Markov. Les contributions suivantes [37, 35, 36] réalisées dans le cadre de cette thèse sont détaillées au chapitre 8.

Thèse de Ouassim Ait El Hara (2012 -) portant sur l'optimisation en grande dimension. Dans ce contexte, une règle de mise à jour du step-size alternative à la mise à jour de CMA-ES pour l'optimisation en grande dimension a été mise au point. L'approche générale suivie utilisant bornes de convergence théoriques et simulations numériques est décrite au chapitre 6. L'article [1] n'est pas décrit en détail mais simplement cité comme illustration de la technique.

Thèse de Asma Atamna (2013 -) qui porte sur l'optimisation sous contrainte et l'amélioration des méthodes d'évaluation d'algorithmes adaptatifs stochastiques. Un article non détaillé dans ce mémoire a été publié [51].

Chapter 3

Introduction in english

This manuscript presents a large part of my research since the end of my PhD. Most of my work is related to numerical (also referred to as continuous) optimization, at the exception of one contribution done during my postdoc in Zurich introducing a new stochastic algorithm to simulate chemical or biochemical systems [23].

The optimization algorithms at the core of my work are adaptive derivative-free stochastic (or randomized) optimization methods. The algorithms are tailored to tackle difficult numerical optimization problems in a so-called black-box context where the objective function to be optimized is seen as a black-box. For a given input solution, the black-box returns solely the objective function value but no gradient or higher order derivatives are assumed. The optimization algorithm can use the information returned by the black-box, i.e. the history of function values associated to the queried search points, but no other knowledge that could be within the black-box (parameters describing the class of functions the function belongs to, ...). This black-box context is very natural in industrial settings where the function to be optimized can be given by an executable file for which the source code is not provided. It is also natural in situations where the function is given by a large simulation code from which it is hard to extract any useful information for the optimization.

This context is also called derivative-free optimization (DFO) in the mathematical optimization community. Well-known DFO methods are the Nelder-Mead algorithm [79, 77], pattern search methods [54, 90, 6] or more recently the NEW Unconstraint Optimization Algorithm (NEWUOA) developed by Powell [82, 81].

In this context, I have been focusing on DFO methods in the literal sense. However the methods my research is centered on have a large stochastic component and originate from the community of bio-inspired algorithms mainly composed of computer scientists and engineers. The methods were introduced at the end of the 70's. A parallel with Darwin's theory of the evolution of species based on blind variation and natural selection was recognized and served as source of inspiration for those methods. Nowadays this field of bio-inspired methods is referred to as *evolutionary computation* (EC) and a generic term for the methods is *evolutionary algorithms*. The probably most famous examples of bio-inspired methods are genetic algorithms (GAs). However today GAs are known to be *not* competitive for *numerical* optimization. Evolution Strategies (ES) introduced in the end of the 70's [83] have emerged as the main sub-branch of EC devoted to continuous optimization. One important feature of ES is that they are comparison-based algorithms. The present most advanced ES algorithm, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [50] is a variable metric method recognized as the state-of-the-art method for stochastic numerical optimization. It is used in many applications in industry and academy.

Because of historical reasons, the developments and work on Evolution Strategies are mainly carried out in the EC field where practice and effectiveness is definitely as (or more) important as having a theorem proven about an algorithm. However ES algorithms are simply adaptive stochastic iterative methods and they need to be studied from a *mathematical* perspective as

well as any other iterative method in optimization or other domain in order to understand the methods better and convince a broader class of people about their soundness. Questions like their convergence and speed of convergence central in optimization need to be addressed.

My research is encompassed within this general context: I am particularly interested by the mathematical aspects of adaptive stochastic methods like ES (and of course CMA-ES) or more generally adaptive stochastic optimization algorithms. Evolution strategies have this attractive facet that while introduced in the bio-inspired and engineering context, they turn out to be methods with deep theoretical foundations related to invariance, information geometry, stochastic approximation and strongly connected to Markov chain Monte Carlo (MCMC) algorithms. Those foundations and connections are relatively new and to a small (for some topics) or large (for others) extent partly related to some of my contributions. They will be explained within the manuscript. I particularly care that the theory I am working on relates to practical algorithms or has an impact on (new) algorithm designs. I attempt to illustrate this within the manuscript.

While optimization is the central theme of my research, I have been tackling various aspect of optimization. Although most of my work is devoted to single-objective optimization, I have also been working on multi-objective optimization where the goal is to optimize simultaneously several conflicting objectives and where instead of a single solution, a set of solutions, the so-called Pareto set composed of the best compromises is searched.

In the field of single-objective optimization, I have been tackling diverse contexts like *noisy* optimization where for a given point in a search space we do not observe one deterministic value but a distribution of possible function values, *large-scale* optimization where one is interested in tackling problems of the order of 10^4 (medium large-scale) to 10^6 variables (large-scale) and to a smaller extent *constrained* optimization.

In addition to investigating theoretical questions, I have been also working on *designing new algorithms* that calls for theory complemented with numerical simulations. Last I have tackled some *applications* mainly in the context of the PhD of Mohamed Jebalia with an application in chromatography and of the PhD of Zyed Bouzarkouna (PhD financed by the French Institute for petrol) on the placement of oil wells.

Furthermore, a non neglect-able part of my research those past years has been devoted to benchmarking of algorithms. Benchmarking complements theory as it is difficult to assess theoretically the performance of algorithms on all typical functions one is interested. The main motivation has then been to improve the standards on how benchmarking is done. Those contributions were done along with the development of the Comparing COntinuous Optimizers platform (COCO).

My work is articulated around three main complementary axis, namely theory / algorithm design and applications. An overview of the contributions presented within this habilitation organized along those axes is given in Figure 3.1.

3.1 Manuscript organization

This manuscript presents mostly some theoretical aspects of comparison-based stochastic continuous black-box optimization related to my own contributions and discusses how the theory presented relates to practice and is useful for (new) algorithm designs. I however also tried to give a general (advanced) introduction to black-box continuous optimization with a (natural) bias towards the methods and concepts I find the most relevant. More specifically:

Chapter 4 is a general introduction to continuous black-box optimization with zero-order methods that presents in particular the state-of-the art CMA-ES algorithm and the connexion between information geometry and black-box optimization. Some definitions introduced within this chapter are useful in the whole manuscript. Chapter 5 is dedicated to invariance in optimization and specifically invariance of comparison-based adaptive methods. It also presents a proof of the affine invariance of CMA-ES. It can also be seen as an introductory chapter. Chapter 6 presents some bounds for specific algorithm frameworks, namely different types of ES algorithms. The relative tightness of the bounds as well as how those bounds are useful for algorithm design is illustrated.

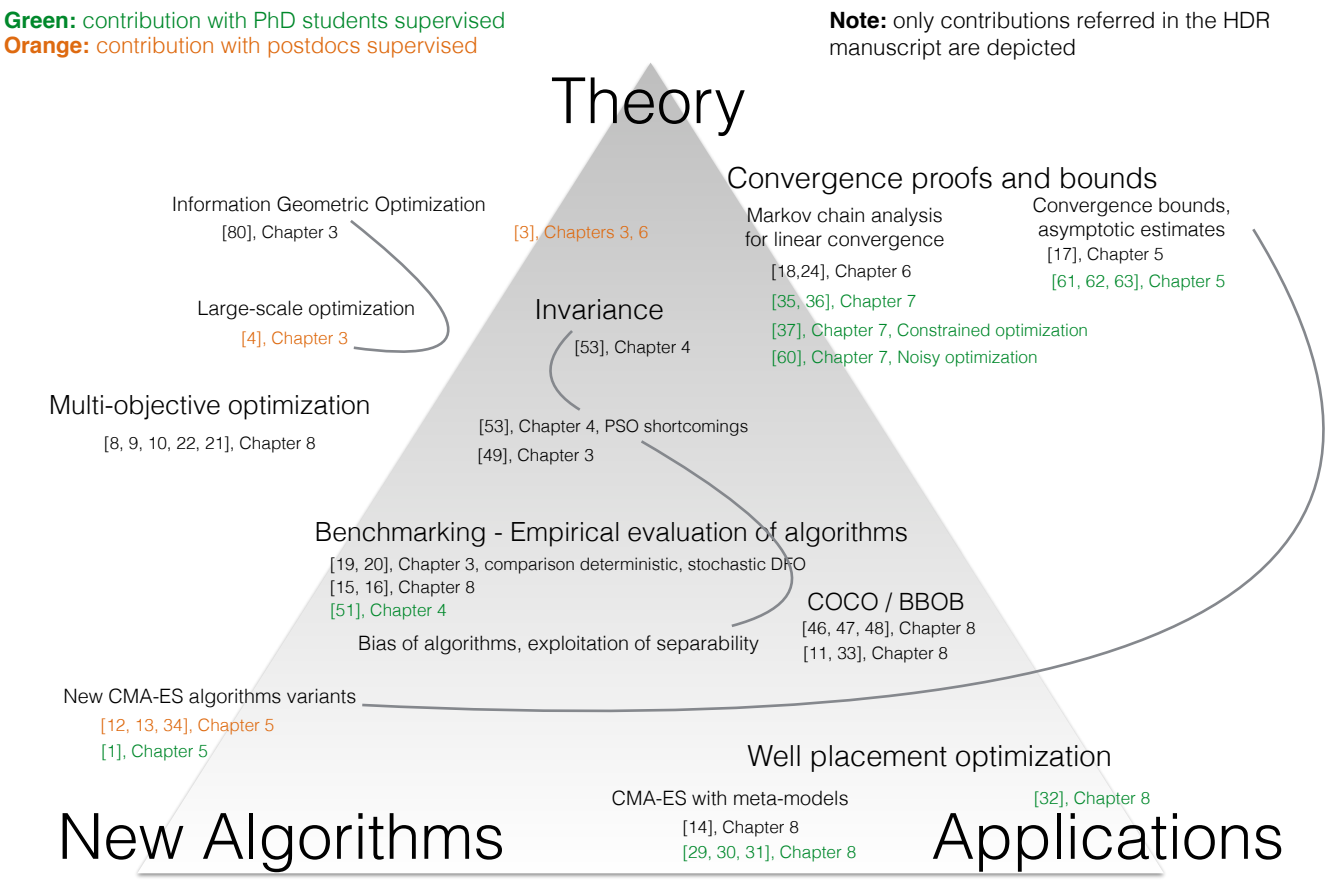

Figure 3.1: Overview of the contributions presented within this habilitation manuscript. In green the contributions with the PhD students supervised or co-supervised, namely Mohamed Jebalia, Zyed Bouzarkouna, Alexandre Chotard, Ouassim AitElhara and Asma Atamna. In orange with the post-doc supervised Dimo Brockhoff and Youhei Akimoto.

In Chapter 7, we present a general methodology that exploits invariance and Markov chain stability analysis for addressing the linear convergence of step-size adaptive algorithms. The chapter makes the connexion between comparison-based adaptive black-box methods and Markov chain Monte Carlo algorithms. In Chapter 8, we present other theoretical results also exploiting the theory of Markov chains to analyze some ES algorithms for constrained and noisy optimization. In Chapter 9, we present other contributions either less theoretical or not related to single-objective optimization, namely on multi-objective optimization, benchmarking and applications.

Note: The references to the papers I am (co)-author appear in cyan, for instance [22] refers to one of my papers while [79] refers to a paper I am not author.

Chapter 4

Adaptive stochastic comparison-based black-box algorithms

Contents

This chapter is intended as a general introduction to black-box optimization with zero order methods. It motivates black-box optimization and the need for stochastic methods in order to approach black-box problems. It introduces the comparison-based feature of the methods investigated within this manuscript. It then gives a formal definition for comparison-based stochastic black-box algorithms that encompasses the state-of-the art CMA-ES that is described in details. This definition is used in the following chapters. We present then the recent finding about the connexion between algorithms like CMA-ES and information geometry and how it opens the way for new algorithm designs in particular in the context of large-scale optimization. We finish by underlying the Markovian and stochastic approximation frameworks behind comparison-based stochastic black-box algorithms. While this chapter is an introduction, the part on the connexion with information geometry is related to [80], the section 4.1 uses results from [20, 19]. How IGO can be used for designing new algorithms in the context of large scale-optimization is related to [4].

4.1 Black-box and derivative-free optimization

Numerical black-box optimization is concerned with the optimization of a function $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ defined on a search space $\mathcal{X}$ subset of $\mathbb{R}^n$ in a so-called black-box scenario. Specifically, the optimization algorithm can query the function at any point $\mathbf{x} \in \mathcal{X}$ of the search space and the black-box function will return the value $f(\mathbf{x})$ only (i.e. zero-order information). The algorithm cannot use any other information about the objective function than the history of queried points together with their function value. This context is very natural when dealing with industrial applications. We will see some examples of applications in Chapter 9.

Methods tailored for black-box (zero-order) optimization are also called derivative-free optimization (DFO) methods, a term introduced within the (mathematical) optimization community that has been developing also methods with derivatives (Newton, quasi-Newton methods, ...). DFO methods have seen a renewed interest in this community those past ten years and an introductory textbook *"Introduction to Derivative-free Optimization"* has been relatively recently published [38]. This textbook is however only centered on *deterministic* DFO like the Nelder-

Mead algorithm [79, 77], pattern search methods [54, 90, 6] or trust-region methods based on interpolating a quadratic model whose most prominent algorithm is the NEWUOA algorithm. This algorithm is also referred to as Powell's method [82, 81, 38] and it is considered as the state-of-the-art deterministic DFO.

4.1.1 Stochastic (comparison-based) black-box algorithms

In this context, the optimization algorithms at the core of this manuscript are stochastic (or randomized) black-box or DFO algorithms. While a formal definition will be given in Section 4.2, we give a simplified description of the method for the time being that does not include all algorithms covered within this manuscript but still encompasses many stochastic DFO and ES algorithms. First of all, a parametrized family of probability distributions $\{P_\theta, \theta \in \Theta\}$ defined on $\mathbb{R}^n$ is given. The distribution P_θ encodes approximately, at a given iteration, the belief about where optimal solutions may lie. The first step of one iteration is to

(i) sample candidate solutions $\mathbf{x}_1, \dots, \mathbf{x}_\lambda$ i.i.d. according to P_θ,

then in a second step

(ii) the candidate solutions are evaluated on f, i.e. $f(\mathbf{x}_1), \dots, f(\mathbf{x}_\lambda)$ are computed

and (iii) the parameter θ is updated using an update function $\theta \leftarrow F\left(\theta, \mathbf{x}_1, \dots, \mathbf{x}_\lambda, f(\mathbf{x}_1), \dots, f(\mathbf{x}_\lambda)\right)$. If the so-defined algorithm works properly, the distribution P_θ will concentrate on local optima of the function f.

An additional feature of many stochastic DFO and of all ES is that the update step (iii) is not using the f-values but the *ranking information* only, i.e. after the evaluation step (ii), a permutation $\mathcal{S}$ containing the ordered solutions is extracted, i.e. $\mathcal{S}$ is such that

$$f(\mathbf{x}_{\mathcal{S}(1)}) \leq f(\mathbf{x}_{\mathcal{S}(2)}) \leq \ldots \leq f(\mathbf{x}_{\mathcal{S}(\lambda)})$$

and the last step consists in

(iii) updating θ according to $\theta \leftarrow G\left(\theta, \mathbf{x}_{\mathcal{S}(1)}, \dots, \mathbf{x}_{\mathcal{S}(\lambda)}\right)$.

The described framework is natural for many ES algorithms but also for so-called estimation of distribution algorithms (EDA) [71]. The ranked-based property is also named *comparison-based* property. We can also talk about *function-value-free* optimization.

On the family of probability distribution $\{P_\theta, \theta \in \Theta\}$ A very common choice for the family of probability distributions $\{P_\theta, \theta \in \Theta\}$ are Gaussian vectors also referred to as multivariate normal distributions in this manuscript. Most Evolution Strategy algorithms use Gaussian distributions[1][2] [27] [45].

A Gaussian distribution is parameterized by a mean vector and a covariance matrix $\mathcal{N}(\mathbf{m}, \mathbf{C})$ where the mean vector $\mathbf{m}$ is generally the favorite or incumbent solution proposed by the algorithm and the matrix $\mathbf{C}$ encodes the geometric shape[3]. However, usually an additional scaling parameter σ is added such that candidate solutions are sampled according to $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$ or equivalently according to

$$\mathbf{m} + \sigma\mathbf{C}^{1/2}\mathcal{N}(0, I_d) \ .$$

[1] Gaussian distributions are convenient for designing algorithms, in particular due to the stability property that the sum of independent normally distributed normal distributions is still a normal distribution. Also, given a fixed mean and standard deviation, a normal distribution has the maximum entropy.

[2] Besides Gaussian distributions, Cauchy mutations have also been used [64, 85, 91], however the main effect observed when using Cauchy distribution is the exploitation of the separability as the sample outcome of coordinate-wise independent Cauchy concentrates along axis [49].

[3] Lines of equal density of a Gaussian vector are hyper-ellipsoids whose equations are of the form $\{\mathbf{x} \in \mathbb{R}^n | (\mathbf{x} - \mathbf{m})^T\mathbf{C}^{-1}(\mathbf{x} - \mathbf{m}) = \text{cst}\}$ such that the eigenvectors of $\mathbf{C}$ represent the axes of the hyper-ellipsoid and the associated eigenvalues encode the elongation along those axis.

In the previous equation we understand that the parameter σ controls the overall scale or the step-size of the steps sampled around the mean $\mathbf{m}$. Given that the covariance matrix has a quadratic number of parameters to adapt, the overall scale is adapted by an independent mechanism that can have a much larger learning rate (i.e. the adaptation can be much faster). In effect, the adaptation of the step-size controls the asymptotic convergence rate of the algorithm or the convergence once the adaptation of the matrix has taken place.

Algorithms adapting the step-size solely are referred to as *step-size adaptive* algorithms (or step-size adaptive evolution strategies or comparison-based step-size adaptive randomized search).

Note that while for multivariate normal distributions, P_θ is entirely characterized by the mean vector and covariance matrix $\sigma^2\mathbf{C}$, often we use θ as a place-holder for the state variables that directly or indirectly relate to the distribution. For instance in CMA-ES the state variables are $\theta = (\mathbf{X}, \sigma, \mathbf{C}, \mathbf{p}, \mathbf{p}^\sigma)$ (see Section 4.2.1) where only $\mathbf{X}, \sigma$ and $\mathbf{C}$ directly encode the Gaussian distribution used for sampling solutions.

Both the stochastic and the comparison-based features confer some robustness. Indeed randomness is naturally a source of robustness as by essence a stochastic algorithm cannot heavily rely on a specific sample outcome (which is by definition uncertain). Hence errors or outliers on the computation of an f-value have only a limited impact on the algorithm performance. In the same vein, if an algorithm uses the ranking of solutions instead of exact function values, outliers or errors have an effect only if they change the ranking of the solutions. Also very small or very large f-values will have a limited impact on the algorithm performance. One major advantage of a comparison-based algorithm is its invariance to composing the objective function to the left by a strictly increasing transformation. It implies that non-convex or non-smooth functions can be as easily optimized as convex ones. This important invariance property will be discussed in Chapter 5.

Stochastic optimization algorithms are sometimes systematically classified as *global optimization* methods in contrast to *local optimization* methods (where typically gradient-based algorithms are local optimization methods). This dichotomy global versus local is unfortunately associated to several downsides:

- a strong emphasis is made on multi-modality (a function is multi-modal if it has several local optima) as if multi-modality was the main source of difficulty a (stochastic) optimization algorithm has to face. Other important sources of difficulties are ignored or neglected (see below),

- a common belief is that stochastic optimization methods do not need to solve efficiently problems where local optimization methods are the method of choice or that it is not important that they converge fast on functions that could be solved by gradient-based methods,

- theoretically, people tend to focus much of their attention on proving convergence to the global optimum (on possibly multi-modal functions) without caring for the *speed* of convergence. Remark that global convergence can be trivial to prove (i.e. the pure random search converges globally). It is a sometimes used (questionable) technique to enforce global convergence of an algorithm by adding an additional step consisting in sampling over the whole search space such that the global optimum can be hit with positive probability (and this probability does not vanish to zero).

Related to the items above, we want to stress that in a black-box scenario, the algorithm does not know in advance the difficulties of the function that needs to be optimized. Often a function combines several difficulties and it is thus important to be able to deal with all of them. We want now to discuss typical difficulties in optimization, that hence a stochastic algorithm should be able to cope with.

4.1.2 What makes a search problem difficult?

Already mentioned, **mutli-modality** is a source of difficulty in optimization but more generally, all type of "**ruggedness**" the function can have can be a source of difficulty. Ruggedness might come from the fact that the function is not differentiable, **not continuous** or can be **noisy** (i.e. two evaluations of the function give different outputs and a distribution of objective function values is observed instead of a single value).

Another source of difficulty is related to **dimensionality**. Due to the curse of dimensionality, a search procedure that can be valid in dimension one or two (like a grid search or a pure random search) is useless or impossible in larger dimensions (i.e. n larger than 4 or 5).

Separability in optimization qualifies functions that satisfy the following property: an objective function $f(x_1, \ldots, x_n)$ is separable if the optimal value for any variable x_i can be obtained by optimizing $f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, x_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n)$ for any fixed choice of the variables $\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n$. Additively decomposable functions, i.e. functions that write $f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}_i)$ are common examples of separable functions. Separable functions can be optimized by n one-dimensional optimization processes. However functions that need to be optimized in practice are usually **non-separable** (otherwise the optimization problem is easy and there is no need for an advanced optimization method).

A last source of difficulty is related to the *conditioning* of a function. For a convex-quadratic function $f(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^\star)^T H(\mathbf{x} - \mathbf{x}^\star)$ where $H \in \mathrm{S}(n, \mathbb{R})$, the conditioning of the function can be defined as the condition number of the matrix H. A large condition number means geometrically a large ratio between the largest and smallest axis length of the ellipsoidal level-set of f[4]. By extension, an **ill-conditioned** function refers to a function with squeezed level sets. Problems are typically considered as ill-conditioned if the conditioning is larger than 10^5. In practice condition numbers up to 10^{10} are frequently encountered[5].

A frequent diagnosis for explaining the failure of an optimization algorithm is *"the algorithm is stuck in a local optimum"*. However often the problem is more related to a high condition number coupled with non-separability of the function that many algorithms cannot solve properly (see for instance the PSO algorithm in the next section).

4.1.3 Performance assessment of stochastic search algorithms on convex and quasi-convex quadratic functions

To finish this section, we present the results of some experiments to quantify the performance of some stochastic search algorithms as *local search algorithms*. The outcome of this experiment might look surprising in the state of mind that global optimization algorithms are poor local search algorithms.

In the papers [20, 19] we have tested three stochastic search algorithms, namely the CMA-ES algorithm, a particle swarm algorithm (PSO)[6] [65] and the differential evolution (DE) algorithm[7] [89]. Those two latter algorithms are quite famous at least in the EC community, this motivated our choice.

[4] Note that the axis ratio is the *square root* of the condition number.

[5] Since the condition number associated to a function has been formally defined for convex-quadratic functions only, we cannot talk about condition number in general. However, the observation of condition number of 10^{10} is related to what is observed with the CMA-ES algorithm: the covariance matrix of CMA learns the inverse of the Hessian matrix on convex-quadratic function, we can then by extension associate the condition number of a function optimized by CMA-ES to the condition number of the inverse of the covariance matrix.

[6] "PSO is a bio-inspired algorithm based on the biological paradigm of a swarm of particles that 'fly' over the objective landscape, exchanging information about the best solutions they have 'seen'. More precisely, each particle updates its velocity, stochastically twisting it toward the direction of the best solutions seen by (i) itself and (ii) some parts of the whole swarm; it then updates its position according to its velocity and computes the new value of the objective function" (sketch of description taken from [20]).

[7] "DE borrows from Evolutionary Algorithms (EAs) the paradigm of an evolving population. However, a specific 'mutation' operator is used that adds to an individual the difference between two others from the population " (sketch of description taken from [20]).

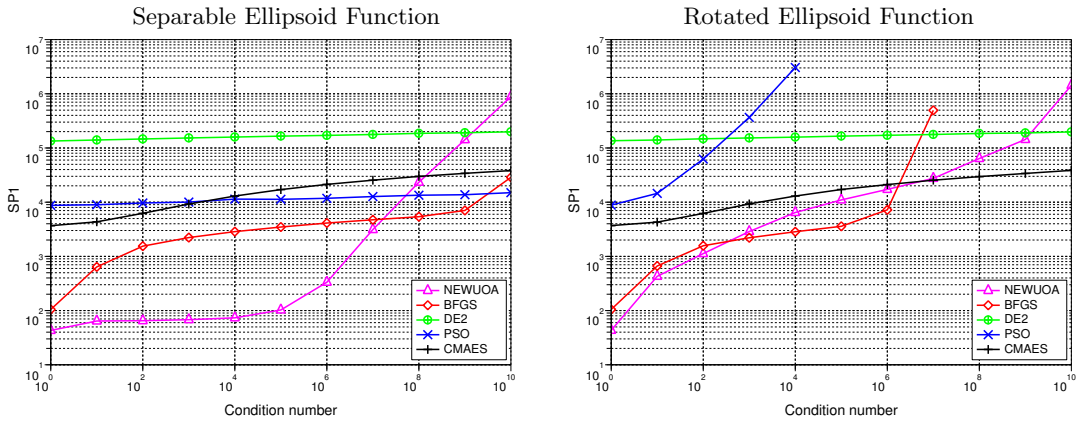

Figure 4.1: Number of function evaluations to reach a target of 10^{-9} on the convex-quadratic functions f_{elli} and f_{ellirot} for five derivative free optimizers. The x-axis depict the condition number of the Hessian matrix of the convex-quadratic function.

In addition we have tested the quasi-Newton algorithm BFGS (used in the derivative free mode, that is the gradients are computed by finite differences) and the NEWUOA algorithm.

The algorithms are tested in the optimal scenario for BFGS and NEWUOA, that is a convex-quadratic function. Note that NEWUOA interpolates a quadratic model. We have tested in particular the impact of the condition number and the separability. We have considered the following test function

$$f_{\text{elli}}(\mathbf{x}) = \sum_{i=1}^{n} \alpha^{\frac{i-1}{n-1}} \mathbf{x}_i^2$$

which condition number equals to α and has a uniform distribution of the eigenvalues of the Hessian matrix in a log-scale. This function being separable, we have tested as well a rotated version of the function, that is $f_{\text{ellirot}}(\mathbf{x}) = f_{\text{elli}}(\mathbf{Rx})$ with $\mathbf{R}$ a (non identity) rotation matrix sampled uniformly in $\text{SO}(n, \mathbb{R})$. We have measured the number of function evaluations to reach a target of 10^{-9} for different condition numbers. The results obtained are reproduced in Figure 4.1.

On the rotated ellipsoid, for condition numbers between 10^2 and 10^5, we observe a factor of 9 between BFGS and CMA-ES and a factor of 2 between NEWUOA and CMA for a condition number of 10^5. For a condition number of 10^7 or higher, CMA-ES outperforms both BFGS[8] and NEWUOA. The DE algorithm is within a factor of ten slower than CMA (larger for small condition numbers, a bit smaller for larger).

We observe that both PSO and NEWUOA are not invariant with respect to a rotation of the problem. While PSO outperforms CMA-ES on the ellipsoid separable function, it cannot solve the non-separable problem with condition number of 10^5 in less than 10^7 function evaluations. We will come back on this aspect in Chapter 5.

We have then measured how the difference in performance is affected when the two previous functions are composed to the left by the strictly increasing transformation $x \mapsto x^{1/4}$, i.e. we have considered the functions $\mathbf{x} \mapsto f_{\text{elli}}(\mathbf{x})^{1/4}$ and $\mathbf{x} \mapsto f_{\text{ellirot}}(\mathbf{x})^{1/4}$, i.e. those functions have the same level sets than $f_{\text{elli}}(\mathbf{x})$ and $f_{\text{ellirot}}(\mathbf{x})$. We observe that the performance of BFGS and NEWUOA degrade while the performance of CMA-ES, PSO and DE remain unchanged due to their invariance to strictly increasing transformations of the objective function (see Chapter 5). We observe very similar performance of BFGS and CMA-ES (within a factor of 2 or less) on the non separable function.

In the end of this manuscript, unless specifically stated, the search space $\mathcal{X}$ is assumed to be equal to $\mathbb{R}^n$.

[8]Note that the decline of the performance of BFGS compared to CMA-ES might be implementation dependent (we used the Matlab toolbox). At least when considering a higher arithmetic precision, results are improved [75].

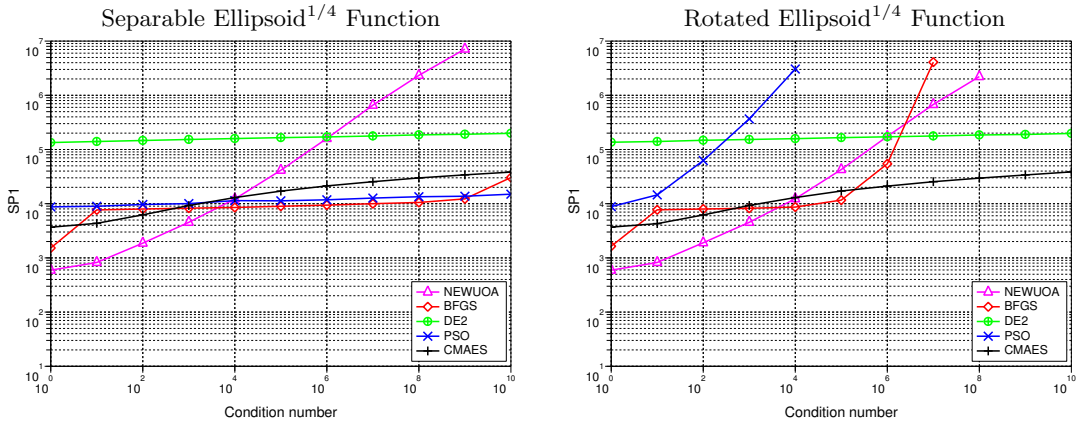

Figure 4.2: Number of function evaluations to reach a target of 10^{-9} on the functions $f_{\text{elli}}^{1/4}$ and $f_{\text{ellirot}}^{1/4}$ for five derivative free optimizers. The x-axis depict the condition number of the Hessian matrix of f_{ellirot}.

4.2 A formal definition of a comparison-based stochastic black-box algorithm

We now introduce a more formal definition of a comparison-based black-box algorithm. We denote by $(\theta_t)_{t\in\mathbb{N}}$ the state variables of the algorithm at iteration t that parametrize P_{θ_t}. Each θ_t belongs to the state space Θ. We consider $(\mathbf{U}_t)_{t\in\mathbb{N}}$ a sequence of i.i.d. random vectors defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ independent of θ_0 and taking values into $\mathbb{R}^{n\lambda}$. We assume that each $\mathbf{U}_t$ has λ coordinates $\mathbf{U}_t = (\mathbf{U}_t^1, \ldots, \mathbf{U}_t^\lambda) \in (\mathbb{R}^n)^\lambda$ (not necessarily i.i.d.) and denote $p_{\mathbf{U}}$ the probability distribution of each $\mathbf{U}_t$. We consider the solution function

$$\mathcal{S}ol : \Theta \times \mathbb{R}^n \to \mathbb{R}^n \tag{4.1}$$

that samples candidate solutions starting from the state θ_t using the random components $\mathbf{U}_{t+1}^i$ such that the new candidate solutions at iteration t read

$$\mathbf{X}_{t+1}^i = \mathcal{S}ol(\theta_t, \mathbf{U}_{t+1}^i), i = 1, \ldots, \lambda \ .$$

We denote $\mathcal{S}$ the permutation of ordered solutions. A comparison-based stochastic black-box algorithm is determined by the data of $(\mathcal{S}ol, p_{\mathbf{U}})$ and an update function $\mathcal{F}$ that updates the state of the algorithm from the ordered coordinates of $\mathbf{U}_{t+1}$.

Definition 1 (Comparison-based stochastic black-box algorithm minimizing $f : \mathbb{R}^n \to \mathbb{R}$)**.** *Let $\lambda \in \mathbb{N}_>$ and $p_{\mathbf{U}}$ be a probability distribution defined on $\mathbb{R}^{n\lambda}$ where each $\mathbf{U}$, distributed according to $p_{\mathbf{U}}$, has a representation $(\mathbf{U}^1, \ldots, \mathbf{U}^\lambda)$ (each $\mathbf{U}^i \in \mathbb{R}^n$). Let $\mathcal{S}ol$ be a mesurable solution function from $\Theta \times \mathbb{R}^n$ to $\mathbb{R}^n$ and $\mathcal{F}$ be an (measurable) update function mapping $\Theta \times \mathbb{R}^{n\lambda}$ onto Θ.*
A comparison-based stochastic black-box algorithm is determined by the triplet $(\mathcal{S}ol, \mathcal{F}, p_{\mathbf{U}})$ from which the recursive sequence $(\theta_t) \in \Theta$ is defined via $\theta_0 \in \Theta$ and for all iteration indexes t:

1. *Sample candidate solutions*
$$\mathbf{X}_{t+1}^i = \mathcal{S}ol(\theta_t, \mathbf{U}_{t+1}^i), i = 1, \ldots, \lambda \ . \tag{4.2}$$
2. *Evaluate the solutions on f, i.e. compute $f(\mathbf{X}_{t+1}^1), \ldots, f(\mathbf{X}_{t+1}^\lambda)$ and rank solutions. Denote $\mathcal{S}$ the permutation containing index of the ordered solutions, i.e.*
$$f\left(\mathbf{X}_{t+1}^{\mathcal{S}(1)}\right) \leq \ldots \leq f\left(\mathbf{X}_{t+1}^{\mathcal{S}(\lambda)}\right) \ . \tag{4.3}$$
3. *Update θ_t:*
$$\theta_{t+1} = \mathcal{F}\left(\theta_t, \mathbf{U}_{t+1}^{\mathcal{S}(1)}, \ldots, \mathbf{U}_{t+1}^{\mathcal{S}(\lambda)}\right) \ . \tag{4.4}$$

where $(\mathbf{U}_t)_{t\in\mathbb{N}_>}$ is an i.i.d. sequence of random vectors on $\mathbb{R}^{n\lambda}$ distributed according to $p_{\mathbf{U}}$.

The previous definition defines a function $\mathcal{G}$ such that the updates write

$$\theta_{t+1} = \mathcal{G}(\theta_t, \mathbf{U}_{t+1}) \tag{4.5}$$

with $(\mathbf{U}_t)_{t\in\mathbb{N}_>}$ i.i.d. More precisely, the function $\mathcal{G}$ equals

$$\mathcal{G}(\theta, \mathbf{u}) = \mathcal{F}(\theta, \mathcal{O}rd(f(\mathcal{S}ol((\theta, \sigma), \mathbf{u}^i))_{i=1,\ldots,\lambda}) * \mathbf{u}) \tag{4.6}$$

where the function $\mathcal{O}rd$ extracts the permutation of ranked solutions and the given a permutation $\mathcal{S}$,

$$\mathcal{S} * \mathbf{u} = \left(\mathbf{u}^{\mathcal{S}(1)}, \ldots, \mathbf{u}^{\mathcal{S}(\lambda)}\right) \ .$$

We present two examples of algorithms following definition 1 namely the CMA-ES algorithm and the (1+1)-ES with one-fifth success rule whose definitions will be needed in other parts of the manuscript.

4.2.1 The CMA-ES algorithm

In CMA-ES, the state of the algorithm is given by $\theta_t = (\mathbf{X}_t, \sigma_t, \mathbf{C}_t, \mathbf{p}_t, \mathbf{p}_t^\sigma) \in (\mathbb{R}^n \times \mathbb{R}^+ \times \mathrm{S}(n, \mathbb{R}) \times \mathbb{R}^n \times \mathbb{R}^n)$. New solutions follow $\mathcal{N}(\mathbf{X}_t, \sigma_t^2\mathbf{C}_t)$, hence only $\mathbf{X}_t$, σ_t and $\mathbf{C}_t$ encode the sampling distribution with $\mathbf{X}_t$ representing the mean of the distribution and $\sigma_t^2\mathbf{C}_t$ its covariance matrix respectively while $\mathbf{p}_t$ and $\mathbf{p}_t^\sigma$ are state variables used to update $\mathbf{C}_t$ and σ_t respectively. The parameter σ_t is the so-called step-size and can be thought of as a global scaling factor for the multivariate normal distribution—this view is not fully accurate as $\mathbf{C}_t$ also plays on the scale since it is not normalized in the algorithm—and has been introduced such that the global scaling can be adapted faster than the covariance matrix. The vectors $\mathbf{p}_t^\sigma$ and $\mathbf{p}_t$ are auxiliary state variables used to update the step-size and the covariance matrix. New solutions follow

$$\mathbf{X}_{t+1}^i = \mathbf{X}_t + \sigma_t \mathbf{C}_t^{1/2} \mathbf{U}_{t+1}^i \text{ where } \mathbf{U}_{t+1}^i \sim \mathcal{N}(0, I_d), i = 1, \ldots, \lambda \ . \tag{4.7}$$

Hence $\mathbf{U}_{t+1} = (\mathbf{U}_{t+1}^1, \ldots, \mathbf{U}_{t+1}^\lambda)$ is a vector of λ standard multivariate normal distributions and formally $\mathcal{S}ol(\theta, \mathbf{U}^i) = [\theta]_1 + [\theta]_2 [\theta]_3^{1/2} \mathbf{U}^i$ where we use the intuitive notation $\theta = ([\theta]_1, [\theta]_2, [\theta]_3, [\theta]_4, [\theta]_5) = (\mathbf{X}, \sigma, \mathbf{C}, \mathbf{p}, \mathbf{p}^\sigma)$.

The λ new solutions are evaluated and ranked according to their f-value, that is

$$f(\underbrace{\mathbf{X}_t + \sigma_t \mathbf{C}_t^{1/2} \mathbf{U}_{t+1}^{1:\lambda}}_{\mathbf{X}_{t+1}^{1:\lambda}}) \leq \ldots \leq f(\underbrace{\mathbf{X}_t + \sigma_t \mathbf{C}_t^{1/2} \mathbf{U}_{t+1}^{\lambda:\lambda}}_{\mathbf{X}_{t+1}^{\lambda:\lambda}}) \tag{4.8}$$

where the ordered indexes are denoted $i:\lambda$ such that the permutation containing the ordered indexes is defined as $\mathcal{S}(1) = 1{:}\lambda, \ldots, \mathcal{S}(\lambda) = \lambda{:}\lambda$. Finally the parameter θ_t is updated, i.e. each component of θ_t is in turn updated. The mean vector $\mathbf{X}_t$ moves towards the μ best solutions (usually $\mu = \lambda/2$)

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \sigma_t \mathbf{C}_t^{1/2} \sum_{i=1}^{\mu} w_i \mathbf{U}_{t+1}^{i:\lambda} = \mathbf{X}_t + \sum_{i=1}^{\mu} w_i (\mathbf{X}_{t+1}^{i:\lambda} - \mathbf{X}_t) \tag{4.9}$$

where w_i are weights summing to one ($\sum w_i = 1$) and typically such that $w_1 \geq w_2 \geq \ldots \geq w_\mu$ (see Section 6.3 for an expression of asymptotic optimal weights, in practice an approximation of those optimal weights is taken). The step-size σ_t is then updated. For this, a vector $\mathbf{p}_t^\sigma$ cumulates iteratively the vectors $\sum_{i=1}^{\mu} w_i \mathbf{U}_{t+1}^{i:\lambda}$ [9]

$$\mathbf{p}_{t+1}^\sigma = (1 - c_\sigma)\mathbf{p}_t^\sigma + \alpha_{c_\sigma} \sum_{i=1}^{\mu} w_i \mathbf{U}_{t+1}^{i:\lambda} \tag{4.10}$$

where the learning rate c_σ belongs to $]0, 1]$ and where $\alpha_{c_\sigma} = \mu_{\text{eff}} \sqrt{c_\sigma(1 - c_\sigma)}$ with $\mu_{\text{eff}} = 1/\sum w_i^2$. The normalization by α_{c_σ} for the rightmost term is such that under random selection, i.e. $\mu_{\text{eff}} \sum_{i=1}^{\mu} w_i \mathbf{U}_{t+1}^{i:\lambda} \sim \mathcal{N}(0, I_d)$, assuming $\mathbf{p}_t^\sigma \sim \mathcal{N}(0, I_d)$ then $\mathbf{p}_{t+1}^\sigma \sim \mathcal{N}(0, I_d)$. The length of the vector $\mathbf{p}_{t+1}^\sigma$ is then compared to the expected length, the vector would have under random selection (hence $E[\|\mathcal{N}(0, I_d)\|]$), and the step-size is increased if this length is larger than under random selection and decreased if it is smaller. The update reads

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_{t+1}^\sigma\|}{E[\|\mathcal{N}(0, I_d)\|]} - 1\right)\right) , \tag{4.11}$$

where d_σ is a so-called damping parameter. Last the covariance matrix is adapted by combining two updates

$$\mathbf{C}_{t+1} = (1 - c_1 - c_\mu)\mathbf{C}_t + c_1 \mathbf{p}_{t+1} \mathbf{p}_{t+1}^T + c_\mu \sum_{i=1}^{\mu} w_i \mathbf{C}_t^{1/2} \mathbf{U}_{t+1}^{i:\lambda} (\mathbf{C}_t^{1/2} \mathbf{U}_{t+1}^{i:\lambda})^T \tag{4.12}$$

$$= (1 - c_1 - c_\mu)\mathbf{C}_t + c_1 \mathbf{p}_{t+1} \mathbf{p}_{t+1}^T + c_\mu \sum_{i=1}^{\mu} w_i \frac{(\mathbf{X}_{t+1}^{i:\lambda} - \mathbf{X}_t)(\mathbf{X}_{t+1}^{i:\lambda} - \mathbf{X}_t)^T}{\sigma_t^2} \tag{4.13}$$

where the vector $\mathbf{p}_t$, similarly to the vector $\mathbf{p}_t^\sigma$, cumulates the steps $\mathbf{C}_t^{1/2} \sum_{i=1}^{\mu} w_i \mathbf{U}_{t+1}^{i:\lambda}$ using proper normalization constants, that is

$$\mathbf{p}_{t+1} = (1 - c)\mathbf{p}_t + \sqrt{c(2 - c)} \mu_{\text{eff}} \mathbf{C}_t^{1/2} \sum_{i=1}^{\mu} w_i \mathbf{U}_{t+1}^{i:\lambda} \ . \tag{4.14}$$

with $c \in]0, 1]$ and $c_1 + c_\mu \in]0, 1]$.

The first update (associated to the coeffecient c_1), the *rank-one* update, adds a rank-one matrix with eigenvector associated to the non-zero eigenvalue equal to $\mathbf{p}_{t+1}$. In other words, the update reinforces the likelihood of steps in the vicinity direction of $\mathbf{p}_{t+1}$. The second update (associated to the coefficient c_2) is the *rank-mu* update and will be discussed in Section 4.3. One specific

[9] Note that the vectors cumulated differ from some normalized steps by the $\mathbf{C}_t^{-1/2}$ multiplication. We will see in Section 5.4 that this seems to be problematic to ensure the affine invariance of the resulting algorithm.

feature of the CMA-ES algorithm is that all parameters are robustly tuned and are not the choice of the user who is finally left to give for starting a run, an initial mean vector $\mathbf{X}_0$ and an initial step-size σ_0. We refer to [45] for the specific default values of the parameters.

We can now give the explicit expression of the update function $\theta_+ = \mathcal{F}(\theta, \mathbf{y})$ of Definition 1

$$[\theta_+]_1 = [\theta]_1 + [\theta]_2 [\theta]_3^{1/2} \sum_{i=1}^{\mu} w_i \mathbf{y}_i \tag{4.15}$$

$$[\theta_+]_2 = [\theta]_2 \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|[\theta_+]_5\|}{E[\|\mathcal{N}(0, I_d)\|]} - 1\right)\right) \tag{4.16}$$

$$[\theta_+]_3 = (1 - c_1 - c_\mu)[\theta]_3 + c_1 [\theta_+]_4 [\theta_+]_4^T + c_\mu \sum_{i=1}^{\mu} w_i [\theta]_3 \mathbf{y}_i ([\theta]_3 \mathbf{y}_i)^T \tag{4.17}$$

$$[\theta_+]_4 = (1 - c)[\theta]_4 + \sqrt{c(2-c)}\mu_{\text{eff}} [\theta]_3^{1/2} \sum_{i=1}^{\mu} w_i \mathbf{y}_i \tag{4.18}$$

$$[\theta_+]_5 = (1 - c_\sigma)[\theta]_5 + \alpha_{c_\sigma} \sum_{i=1}^{\mu} w_i \mathbf{y}_i \tag{4.19}$$

where $\theta = ([\theta]_1, [\theta]_2, [\theta]_3, [\theta]_4, [\theta]_5) = (\mathbf{X}, \sigma, \mathbf{C}, \mathbf{p}, \mathbf{p}^\sigma)$.

4.2.2 The (1+1)-ES with one-fifth success rule

The so-called $(1 + 1)$-ES with one-fifth success rule is one of the oldest step-size adaptive randomized algorithms [41, 86, 83]. The algorithm samples new solutions with a multivariate normal distribution having a covariance matrix proportional to the identity (i.e. isotropic samplings). The mean of the distribution and its scaling are adjusted with the parameters denoted $\mathbf{X}_t$ and σ_t respectively. Following the notations of Definition 1, $\theta_t = (\mathbf{X}_t, \sigma_t)$. At each iteration t, a new candidate solution following $\mathcal{N}(\mathbf{X}_t, \sigma_t^2 I_d)$ is sampled, that is

$$\mathbf{X}_{t+1}^1 = \mathbf{X}_t + \sigma_t \mathcal{N}_{t+1} \ . \tag{4.20}$$

where $\mathcal{N}_{t+1} \sim \mathcal{N}(0, I_d)$. After the computation of $f(\mathbf{X}_{t+1}^1)$, the value obtained is compared to $f(\mathbf{X}_t)$, and the new mean equals the best solution among $\mathbf{X}_{t+1}^1$ and $\mathbf{X}_t$, i.e.

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \sigma_t \mathcal{N}_{t+1} 1_{\{f(\mathbf{X}_{t+1}^1) \leq f(\mathbf{X}_t)\}} \ . \tag{4.21}$$

To comply to Definition 1, the solution operator is defined as

$$\mathcal{S}ol(\theta, \mathbf{U}^i) = [\theta]_1 + [\theta]_2 \mathbf{U}^i$$

where $\mathbf{U} = (\mathbf{U}^1, \mathbf{U}^2)$ with $\mathbf{U}^1$ following $\mathcal{N}(0, I_d)$ and $\mathbf{U}^2 = 0$ (i.e. dirac-delta distribution in zero).

The step-size is then increased in case of success and decreased otherwise, i.e.

$$\sigma_{t+1} = \sigma_t \left((\gamma - \gamma^{-1/4}) 1_{\{f(\mathbf{X}_{t+1}^1) \leq f(\mathbf{X}_t)\}} + \gamma^{-1/4}\right) \tag{4.22}$$

with $\gamma > 1$. The coefficient $\gamma^{-1/4}$ is such that the step-size is unbiased if the probability of success equals $1/5$. The probability of success $1/5$ is considered as "optimal" (see Section 6.4).

For the $(1 + 1)$-ES with $1/5$ success rule, the update function $\theta_+ = \mathcal{F}(\theta, \mathbf{y})$ of Definition 1 is given by

$$[\theta_+]_1 = [\theta]_1 + [\theta]_2 \mathbf{y}_1 \tag{4.23}$$

$$[\theta_+]_2 = [\theta]_2 \left((\gamma - \gamma^{-1/4}) 1_{\{\mathbf{y}_1 \neq 0\}} + \gamma^{-1/4}\right) \ . \tag{4.24}$$

This algorithm is an example of an elitist algorithm, i.e. the best solution is preserved from one iteration to the next one. It cannot be described with the framework presented in Section 4.1.1 because the two candidate solutions compared at a given iteration do not derive from the same sampling distribution.

4.3 An information geometry perspective

An algorithm like CMA-ES has some connections with information geometry and gradient optimization on the Riemannian manifold formed by the family $\{P_\theta, \theta \in \Theta\}$. The understanding of those links is recent: the two first publications in that direction being [2, 43] while [80] refines the connections and present them in a broader context introducing the so-called Information Geometric Optimization (IGO). We explain here briefly the main aspects of the IGO framework and how its instanciation on the family of Gaussian distribution recovers the CMA-ES with rank-mu update.

4.3.1 Defining a joint criterion on the manifold of the family of probability distributions

We have presented a simple framework in Section 4.1.1 to cast some stochastic (comparison-based) black-box algorithms. Given a family of probability distributions $\{P_\theta, \theta \in \Theta\}$, the algorithm is iteratively updating θ parametrizing the family P_θ that encodes the belief of where optimal solutions may be located. Hence, while the optimization is taking place on $\mathbb{R}^n$, the algorithm operates on P_θ.

We assume that $\{P_\theta, \theta \in \Theta\}$ is a Riemannian manifold. We explain now how *some* θ-updates can be framed as a gradient update step on Θ. For this, a joint criterion to be optimized on $\{P_\theta, \theta \in \Theta\}$ needs to be defined first. Given that we want to minimize, $f : \mathbb{R}^n \to \mathbb{R}$, one easy way to construct a criterion on Θ is to consider the expected value of f under P_θ

$$\tilde{J}(\theta) = \int f(\mathbf{x}) P_\theta(d\mathbf{x}) \ . \tag{4.25}$$

This criterion is such that minimizing $\tilde{J}$ means finding P_θ concentrated on the global minimum of f [2, 43]. However, $\tilde{J}$ is not invariant to strictly increasing transformations of f and does not lead to *comparison-based* algorithms. A more complex criterion on Θ needs thus to be considered.

Given θ_t the current parameter value, define on Θ the joint criterion

$$J_{\theta_t}(\theta) = \int W^f_{\theta_t}(\mathbf{x}) P_\theta(d\mathbf{x}) \tag{4.26}$$

where $W^f_{\theta_t}$ is a monotone rewriting of f that depends on θ_t and that is defined below. Define first $q^{\leq}_\theta(\mathbf{x})$ as the probability to sample better values than $f(\mathbf{x})$, i.e.

$$q^{\leq}_\theta(\mathbf{x}) = \Pr_{\mathbf{x}' \sim P_\theta} \left(f(\mathbf{x}') \leq f(\mathbf{x}) \right)$$

and $q^{<}_\theta(\mathbf{x})$ as the probability to sample strictly better values than $f(\mathbf{x})$, i.e.

$$q^{<}_\theta(\mathbf{x}) = \Pr_{\mathbf{x}' \sim P_\theta} \left(f(\mathbf{x}') < f(\mathbf{x}) \right) \ .$$

Let $w : [0, 1] \to \mathbb{R}$ be a non-increasing function that represents the selection scheme. The monotone rewriting of f is defined as the following function of $q^{\leq}_\theta$ and $q^{<}_\theta(\mathbf{x})$

$$W^f_\theta(\mathbf{x}) = \begin{cases} w(q^{\leq}_\theta(\mathbf{x})) & \text{if } q^{\leq}_\theta(\mathbf{x}) = q^{<}_\theta(\mathbf{x}) \\ \frac{1}{q^{\leq}_\theta(\mathbf{x}) - q^{<}_\theta(\mathbf{x})} \int_{q=q^{<}_\theta(\mathbf{x})}^{q^{\leq}_\theta(\mathbf{x})} w(q) dq & \text{otherwise} \, . \end{cases} \tag{4.27}$$

The definition of W^f_θ is invariant under strictly increasing transformations of f. For any θ_t, if P_θ is concentrated on the global minimum of f, then $\theta \mapsto J_{\theta_t}(\theta)$ is maximal equal to $w(0)$. The objective is hence to *maximize* $\theta \mapsto J_{\theta_t}(\theta)$, the expected value of $W^f_{\theta_t}(\mathbf{x})$ over P_θ.

4.3.2 (Natural) Gradient ascent on the joint criterion

A gradient step can be performed to maximize J_{θ_t}. However, the gradient should be taken with respect to the Fisher metric such that the resulting gradient is invariant with respect to the chosen θ-parametrization [80]. This latter gradient is referred to as "natural" gradient and denoted $\widetilde{\nabla}_\theta J_{\theta_t}(\theta)$ while the gradient composed of the partial derivatives w.r.t. the given parametrization is called vanilla gradient.

A gradient ascent step to maximize J_{θ_t} reads

$$\theta_{t+\delta t} = \theta_t + \delta t \widetilde{\nabla}_\theta J_{\theta_t}(\theta)|_{\theta=\theta_t} = \theta_t + \delta t \left(\widetilde{\nabla}_\theta \int W^f_{\theta_t}(\mathbf{x}) P_\theta(d\mathbf{x}) \right) \Big|_{\theta=\theta_t} , \tag{4.28}$$

where δt is a time increment or step-size of the gradient step. The natural gradient of the integral in the previous equation can be rewritten as

$$\widetilde{\nabla}_\theta \int W^f_{\theta_t}(\mathbf{x}) P_\theta(d\mathbf{x}) = \int W^f_{\theta_t}(\mathbf{x}) \widetilde{\nabla}_\theta \ln P_\theta(\mathbf{x}) P_\theta(d\mathbf{x}) \tag{4.29}$$

such that a new expression for (4.28) reads

$$\theta_{t+\delta t} = \theta_t + \delta t \int W^f_{\theta_t}(\mathbf{x}) \left. \widetilde{\nabla}_\theta \ln P_\theta(\mathbf{x}) \right|_{\theta=\theta_t} P_{\theta_t}(d\mathbf{x}) \ . \tag{4.30}$$

The natural gradient of the log-likelihood $\widetilde{\nabla}_\theta \ln P_\theta(\mathbf{x})$ can be expressed using the Fisher information matrix $I(\theta)$ via

$$\widetilde{\nabla}_\theta \ln P_\theta(\mathbf{x}) = I^{-1}(\theta) \frac{\partial \ln P_\theta(\mathbf{x})}{\partial \theta}$$

where the Fisher information matrix is defined by

$$I_{ij}(\theta) = \int_{\mathbf{x}} \frac{\partial \ln P_\theta(\mathbf{x})}{\partial \theta_i} \frac{\partial \ln P_\theta(\mathbf{x})}{\partial \theta_j} P_\theta(d\mathbf{x}) \ .$$

Overall, the update of θ_t reads

$$\theta_{t+\delta t} = \theta_t + \delta t I^{-1}(\theta_t) \int W^f_{\theta_t}(\mathbf{x}) \left. \frac{\partial \ln P_\theta(\mathbf{x})}{\partial \theta} \right|_{\theta=\theta_t} P_{\theta_t}(d\mathbf{x}) \ . \tag{4.31}$$

Remark that this update does not depend on the gradient of f. The update of θ_t in (4.31) does not yet lead to a tractable algorithm as the integral cannot be computed exactly in general. However a Monte-Carlo approximation of the integral can be performed and lead to the so-called Information Geometric Optimization (IGO) algorithm as explained in the next section.

4.3.3 Monte-Carlo approximation of the gradient of the joint criterion: the IGO algorithm

In order to approximate the integral in (4.31), at each time step, we draw λ samples $\mathbf{x}_1, \dots, \mathbf{x}_\lambda$ according to P_{θ_t} (we assume we have no ties). To approximate $w(\Pr_{\mathbf{x}' \sim P_{\theta_t}}(f(\mathbf{x}') < f(\mathbf{x}_i)))$, we rank the samples according to f and define $\mathrm{rk}(\mathbf{x}_i) = \#\{j | f(\mathbf{x}_j) < f(\mathbf{x}_i)\}$, then

$$w(\Pr_{\mathbf{x}' \sim P_{\theta_t}}(f(\mathbf{x}') < f(\mathbf{x}_i))) \approx w\left(\frac{\mathrm{rk}(\mathbf{x}_i) + \frac{1}{2}}{\lambda} \right)$$

and the update of θ_t stemming from (4.31) with the integral approximated via Monte-Carlo reads

$$\theta_{t+\delta t} = \theta_t + \delta t I^{-1}(\theta_t) \frac{1}{\lambda} \sum_{i=1}^{\lambda} w\left(\frac{\mathrm{rk}(\mathbf{x}_i) + \frac{1}{2}}{\lambda} \right) \left. \frac{\partial \ln P_\theta(\mathbf{x}_i)}{\partial \theta} \right|_{\theta=\theta_t} . \tag{4.32}$$

The estimator of the integral $\int W^f_{\theta_t}(\mathbf{x}) \left.\frac{\partial \ln P_\theta(\mathbf{x})}{\partial \theta}\right|_{\theta=\theta_t} P_{\theta_t}(d\mathbf{x})$ used in the previous equation is consistent (see Theorem 6 in [80]) but biased in general. An other equivalent way to write the update (4.32) is by setting

$$w_i = \frac{w((i-1/2)/\lambda)}{\lambda} \tag{4.33}$$

and denoting $\mathbf{x}_{i:\lambda}$, the i^{th} sampled point ranked according to f, that is the points $\mathbf{x}_{i:\lambda}$ satisfy

$$f(\mathbf{x}_{1:\lambda}) < \ldots < f(\mathbf{x}_{\lambda:\lambda})$$

(assuming we have no ties). The equivalent expression to (4.32) then reads

$$\theta_{t+\delta t} = \theta_t + \delta t I^{-1}(\theta_t) \sum_{i=1}^{\lambda} w_i \left.\frac{\partial \ln P_\theta(\mathbf{x}_{i:\lambda})}{\partial \theta}\right|_{\theta=\theta_t} . \tag{4.34}$$

Both updates are referred to as the Information Geometric Optimization (IGO) algorithm.

4.3.4 Recovering part of CMA-ES

In the case of multivariate normal distributions, the Fisher information matrix and its inverse are known such that the IGO algorithm update can be made more explicit. Interestingly, it recovers then the rank-mu update part of CMA-ES. More precisely let us denote $\theta_t = (\mathbf{X}_t, \mathbf{C}_t)$ with $\mathbf{X}_t$ the mean and $\mathbf{C}_t$ the covariance matrix of the multivariate normal distribution at iteration t, (4.34) simplifies to

$$\mathbf{X}_{t+\delta t} = \mathbf{X}_t + \delta t \sum_{i=1}^{\lambda} w_i (\mathbf{x}_{i:\lambda} - \mathbf{X}_t) \tag{4.35}$$

$$\mathbf{C}_{t+\delta t} = \mathbf{C}_t + \delta t \sum_{i=1}^{\lambda} w_i \left((\mathbf{x}_{i:\lambda} - \mathbf{X}_t)(\mathbf{x}_{i:\lambda} - \mathbf{X}_t)^T - \mathbf{C}_t \right) . \tag{4.36}$$

Those updates coincide with CMA-ES with rank-mu update (i.e. setting $c_1 = 0$, $c_\sigma = 0$ and $\sigma_0 = 1$) and where a learning rate equal to c_μ is added to the update of the mean vector. Note that the weights in (4.35) are not exactly the weights used in the description of CMA-ES in Section 4.2.1. However, denoting to avoid confusion the weights used in Section 4.2.1 w_i^{CMA}, then $w_i^{\text{CMA}} = w_i / \sum_{i=1}^{\lambda} w_i$ and $c_\mu = \delta t \sum_{i=1}^{\lambda} w_i$.

Using an exponential parametrization for the covariance matrix, the IGO algorithm recovers the xNES algorithm [43].

4.3.5 The IGO flow

The IGO flow is the set of continuous-time trajectories in space Θ underlying the IGO algorithm and defined by the ordinary differential equation

$$\frac{d\theta_t}{dt} = \left(\widetilde{\nabla}_\theta \int W^f_{\theta_t}(\mathbf{x}) P_\theta(d\mathbf{x}) \right)\Big|_{\theta=\theta_t} \tag{4.37}$$

$$= \int W^f_{\theta_t}(\mathbf{x}) \left. \widetilde{\nabla}_\theta \ln P_\theta(\mathbf{x}) \right|_{\theta=\theta_t} P_{\theta_t}(d\mathbf{x}) \tag{4.38}$$

$$= I^{-1}(\theta_t) \int W^f_{\theta_t}(\mathbf{x}) \left.\frac{\partial \ln P_\theta(\mathbf{x})}{\partial \theta}\right|_{\theta=\theta_t} P_{\theta_t}(d\mathbf{x}) . \tag{4.39}$$

Hence the IGO algorithm results from a time-discretization and Monte-Carlo approximation of the equation defining the IGO flow.

The convergence of the IGO flow for a standard multivariate normal distribution with a covariance matrix equal to $\sigma_t I_d$ has been studied in [3] where we prove on monotonic C^2 composite functions having positive definite Hessian at critical points of the function, the local convergence of the flow solutions towards the critical points. This holds under the assumptions that (i) w is non-increasing and Lipschitz continuous with $w(0) > w(1)$ and that (ii) the standard deviation σ_t diverges exponentially on a linear function.

Remark on IGO

We have deliberately described the IGO framework in the context of continuous optimization, however the definition of IGO does not exploit the continuous structure of the search space. It has been actually defined on an arbitrary search space [80]. Interestingly, in the case of a discrete search space equal to $\{0,1\}^n$, an other famous algorithm is recovered by the IGO framework, namely the Population Based Incremental Learning algorithm [25].

4.3.6 Large-scale optimization using IGO

The connexion between CMA-ES and IGO opens the ways for designing new algorithms. For instance, in large-scale optimization where one is interested to optimize functions with more than 10^2 or 10^3 variables, it becomes too expensive to use the full CMA-ES algorithm where n^2 parameters in the the covariance matrix need to be learned. Instead, it seems promising to consider parametrization of the covariance matrix with a linear number of parameters. We have proposed such an algorithm and derived the covariance matrix update from the IGO equation. We have then coupled this covariance matrix update with the cumulation concepts used in CMA-ES [4].

4.4 Markovian and stochastic approximation models

To conclude this introductory chapter, we want to stress two mathematical models that can be used to analyze the convergence of comparison-based stochastic algorithms and that naturally follow from the presentation in the previous sections.

On the one hand, comparison-based black-box algorithms have a Markovian structure. From the definition 1 follows that the θ_t update can be written as

$$\theta_{t+1} = \mathcal{G}(\theta_t, \mathbf{U}_{t+1})$$

where $\mathcal{G}$ is the composition of the update function $\mathcal{F}$ and the ordering of the candidate solutions (see (4.5) and (4.6)). Hence θ_{t+1} is a deterministic function of θ_t and an independent vector $\mathbf{U}_{t+1}$, which is a typical form for a Markov chain following a non-linear state-space model [78]. This Markovian structure is heavily exploited to study the convergence of some comparison-based stochastic algorithms in several of my contributions that will be detailed in Chapter 7 and Chapter 8.

On the other hand, the presentation of the IGO algorithm as a time discretization and Monte-Carlo approximation of the equation defining the IGO flow suggests to use the ODE method or stochastic approximation framework to study the convergence of comparison-based algorithms [74, 28, 69]. More precisely with the stochastic approximation framework, recursions of the form

$$\theta_{t+1} = \theta_t + \delta t F_t = \theta_t + \delta t (F(\theta_t) + M_t)$$

are studied where F is the so-called mean field defined as $F(\theta) = E[F_t | \theta_t = \theta]$ and $M_t = F_t - F(\theta_t)$ is a martingale difference sequence. The idea is then to control the error between the stochastic process θ_t and the solution of the ODE $d\theta/dt = F(\theta)$ assuming decreasing step-sizes δt or small enough δt. For instance in the case of the IGO update (4.32),

$$F_t = I^{-1}(\theta_t) \frac{1}{\lambda} \sum_{i=1}^{\lambda} w\left(\frac{\mathrm{rk}(\mathbf{x}_i) + \frac{1}{2}}{\lambda}\right) \left.\frac{\partial \ln P_\theta(\mathbf{x}_i)}{\partial \theta}\right|_{\theta=\theta_t} .$$

Note that the mean field does not coincide in general with the RHS of (4.37) because the estimator in IGO is biased.

Chapter 5

Invariance

Contents

This chapter is a general chapter about invariance in optimization. Some definitions introduced here will be central for Chapter 7 and more generally underlying to many arguments of some theoretical results presented in the manuscript. The definitions given are the synthesis and slight generalization of several articles, namely [52, 24, 53]. While invariance is a general concept in science, we never found in the optimization literature invariance definitions suitable for our purpose. Indeed when the state of the algorithm is reduced to a single vector of $\mathbb{R}^n$ and the algorithm is deterministic, definitions are much simpler (and often so trivial that they are not even formalized) [40]. In our case, we however have to deal with (i) stochastic algorithms that in addition (ii) have usually more state variables than just the estimate of the solution of the optimization problem. We also decided to present a proof of the affine-invariance of CMA given the importance of the result[1]. The formalization of the proof is new. As we will see, it is not completely trivial as it requires a modification in the default CMA algorithm. It however shows that CMA shares the highly desirable affine-invariance property together with Newton and Nelder-Mead algorithms.

[1]This result is not yet published but should hopefully be submitted in the coming months.

5.1 Introduction

Invariance is an important general concept in science related to generalization of results. In short, an algorithm is invariant if it does not change its behavior under the exchange of f with a function in an associated equivalent class, however in general, conditionally to an appropriate change of initial state. Invariance is relevant in that if an algorithm exhibits a certain type of invariance, its performance on *a specific* function can be *generalized* to the *whole* class of functions associated to the invariance class. Invariance is particularly important in a *black-box* scenario where we do not know in advance the properties of the function to be optimized.

A simple invariant in optimization is translation invariance which is usually taken for granted. Roughly speaking, a translation invariant algorithm will behave the same on $\mathbf{x} \mapsto f(\mathbf{x})$ or on $\mathbf{x} \mapsto f(\mathbf{x} - \mathbf{x}_0)$ for all $\mathbf{x}_0 \in \mathbb{R}^n$ and more precisely it will exhibit "translated" traces (i.e. the translated sequence of candidate solutions) on both functions given that the initial states are also properly translated.

We start now by giving a general definition of invariance. We consider the set $\mathfrak{F}$ of all objective functions mapping $\mathbb{R}^n$ to $\mathbb{R}$

$$\mathfrak{F} = \{f : \mathbb{R}^n \to \mathbb{R}\} \ . \tag{5.1}$$

We define $\mathcal{H}$ a mapping that associates to each $f \in \mathfrak{F}$, a function class, which is just a subset of $\mathfrak{F}$. Hence $\mathcal{H}$ is a mapping from $\mathfrak{F}$ to the power set of $\mathfrak{F}$:

$$\mathcal{H} : \mathfrak{F} \to 2^{\mathfrak{F}} \ .$$

We consider for the moment, in order to simplify, a deterministic algorithm and more precisely the mapping it induces on the state space Θ between two iterations, i.e. $\mathcal{A} : \Theta \to \Theta$. When optimizing f the update of the state θ_t is given for all t by

$$\theta_{t+1} = \mathcal{A}^f(\theta_t) \ ,$$

where we add the function optimized as superscript to $\mathcal{A}$.

Definition 2. *([52]) Let $\mathcal{H} : \mathfrak{F} \to 2^{\mathfrak{F}}$ that maps a function $f \in \mathfrak{F}$ to a set of functions (thought as equivalent class of f). Then the algorithm $\mathcal{A}$ is* **invariant under** *$\mathcal{H}$ if for all f, for all $h \in \mathcal{H}(f)$, there exists a bijective state-space transformation $T_{f\to h} : \Theta \to \Theta$ such that for all state $\theta \in \Theta$*

$$\mathcal{A}^f(\theta) = T_{f\to h}^{-1} \circ \mathcal{A}^h \circ T_{f\to h}(\theta) \ . \tag{5.2}$$

If $T_{f\to h}$ is the identity for all f, h then the algorithm $\mathcal{A}$ is **unconditionally invariant**. Equation (5.2) reveals that $\mathcal{A}$ would optimize h just like f if one would be able to perform first the correct change of variables for the initial state $T_{f\to h}$. If we are not in the unconditional invariance scenario, we will typically observe an *adaptation phase* that can be thought of as the time needed to forget the initial state. After this adaptation stage the algorithms will perform equivalently. We will illustrate this point later on when discussing the affine invariance of CMA-ES.

We consider next invariance to monotonically increasing transformations of f (Section 5.2) and then invariance in the search space where the invariance class associated to each f is coming from a group action (Section 5.3). As specific cases we will detail, translation, scale, rotation and affine invariances.

5.2 Invariance to monotonically increasing transformations of comparison-based algorithms

We consider now comparison-based algorithms as defined in Definition 1. Invariance to monotonically increasing transformations is then quite straightforward. Formally let us define $\mathcal{M}_I$ the set of strictly increasing functions $g : I \to \mathbb{R}$, where I is a subset of $\mathbb{R}$, i.e. if for all x and y in I such that $x < y$ we have $g(x) < g(y)$ and define $\mathcal{M} = \cup_I \mathcal{M}_I$. Given any $f : \mathbb{R}^n \to \mathbb{R}$ and any

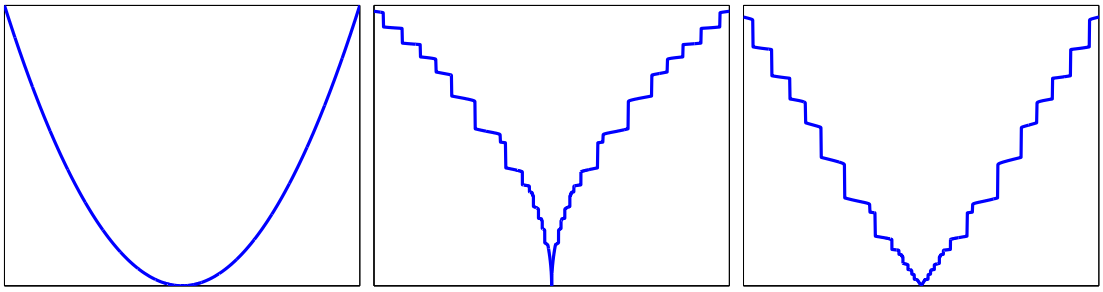

Figure 5.1: Illustration of invariance to strictly increasing transformations. Representation of three instances of functions belonging to the invariance (w.r.t. strictly increasing transformations) class of $f(\mathbf{x}) = \|\mathbf{x}\|^2$ in dimension 1. On the left the sphere function and middle and right functions $g \circ f$ for two different $g \in \mathcal{M}$.

$g \in \mathcal{M}_{f(\mathbb{R}^n)}$, the ranking of candidate solutions being the same on f or $g \circ f$, a comparison-based stochastic black-box algorithm executed on f or on $g \circ f$ will exhibit the same sequence θ_t provided the same initial states and same random sequence $(\mathbf{U}_t)_{t \in \mathbb{N}_>}$ are taken when running on f or $g \circ f$. Let us define the mapping

$$\mathcal{H} : f \in \mathfrak{F} \to \{g \circ f \,|\, g \in \mathcal{M}_{f(\mathbb{R}^n)}\} \ .$$

Then, a comparison-based stochastic black-box algorithm is *unconditionally invariant* with respect to $\mathcal{H}$ or invariant with respect to monotonically increasing transformations.

In Figure 5.1, we depict three elements of $\mathcal{H}(f)$ where $f(\mathbf{x}) = \|\mathbf{x}\|^2$ in dimension 1. Hence, the same trace and thus same convergence rate will be observed for comparison-based algorithms.

This invariance property is shared by pattern search [90, 72] and Nelder-Mead methods [79, 70]. Note that a particular case of strictly increasing functions are affine functions: $\mathbf{x} \in \mathbb{R} \mapsto \alpha\mathbf{x} + \beta$ with $\alpha > 0$. Thus comparison-based algorithms are *affine covariant* (affine covariance relates to affine transformations of f, i.e. by composing f to the left by an affine transformation of $\mathbb{R}$, while affine invariance to affine transformation of the search space, i.e. composition by an affine transformation to the right of f, see [40]).

5.3 Invariance in the search space via group actions

We consider invariance in the search space where transformations are stemming from group actions. In this case, a finer definition of invariance that implies Definition 2 can be given. This new definition comprises affine-invariance, scale-invariance, rotational invariance and translation invariance. We consider as a first example of transformations on $\mathfrak{F}$ the translation. For a function $f \in \mathfrak{F}$, we denote $f_{\mathbf{x}_0}$ the function $f_{\mathbf{x}_0}(\mathbf{x}) = f(\mathbf{x} - \mathbf{x}_0)$. This new function is induced by the action of the group $(\mathbb{R}^n, +)$ on $\mathfrak{F}$ defined as $(\mathbf{x}_0, f) \to f_{\mathbf{x}_0}$.

The notion of translation invariance for an algorithm should reflect that the algorithm "optimizes in the same way" the function f and $f_{\mathbf{x}_0}$ for all possible f and for all possible $\mathbf{x}_0$. For translation invariance, it is intuitive that the algorithm should satisfy that $\mathbf{X}'_t = \mathbf{X}_t + \mathbf{x}_0$ where $\mathbf{X}_t$ is the candidate solution proposed by the algorithm optimizing f and $\mathbf{X}'_t$ the candidate solution proposed when optimizing $f_{\mathbf{x}_0}$ (this requires that we set initial conditions such that $\mathbf{X}'_0 = \mathbf{X}_0 + \mathbf{x}_0$). More precisely, assume that the state of the algorithm considered is reduced to $\mathbf{X}_t$, the algorithm is translation invariant if for all $\mathbf{x}_0$ there exists a bijective transformation of the state of the algorithm, $T_{\mathbf{x}_0} : \mathbf{X} \in \Theta \to \mathbf{X} + \mathbf{x}_0 \in \Theta$ such that $\mathcal{A}^f = T_{\mathbf{x}_0}^{-1} \circ \mathcal{A}^{f_{\mathbf{x}_0}} \circ T_{\mathbf{x}_0}$ for all f. We remark here that the bijective transformation of the state depends on $\mathbf{x}_0$ but not on f in contrast to Definition 2.

More generally, let us consider a group $(G, *)$ (where id_G denotes its neutral element) that

acts on the set of functions $\mathfrak{F}$ and let us denote f_g a transformed function via the action of G:

$$(G, *) \times \mathfrak{F} \mapsto \mathfrak{F} \tag{5.3}$$

$$(g, f) \mapsto g.f := (\mathbf{x} \to f_g(\mathbf{x})) \tag{5.4}$$

Remark that from the group action property $f_{id_G} = f$.

Definition 3. *An algorithm $\mathcal{A}$ is invariant with respect to the search space transformations induced by a group action if for all $g \in G$, there exists a bijective state space transformation T_g such that*

$$T_g^{-1} \mathcal{A}^{f_g} T_g = \mathcal{A}^f \tag{5.5}$$

holds for all $f \in \mathfrak{F}$.

This latter definition implies Definition 2. This definition reads that the sequence of states of the algorithm $\mathcal{A}$ optimizing f and f_g is connected via the change of variables given by T_g. From the previous definition and from the group action property on $\mathfrak{F}$ follow that for all g_1 and g_2 in G

$$T_{g_2}^{-1} \mathcal{A}^{f_{g_2 * g_1}} T_{g_2} = \mathcal{A}^{f_{g_1}} \quad . \tag{5.6}$$

This definition turns out to be equivalent to the following one (we have included a small proof of this statement in the appendix).

Definition 4. *An algorithm $\mathcal{A}$ is invariant with respect to the search space transformations induced by a group action if there exists a group homomorphism $\Phi : (G, *) \to (S_\Theta, \circ)$ where S_Θ is the group of all bijective state space transformations on Θ such that*

$$\Phi_g^{-1} \mathcal{A}^{f_g} \Phi_g = \mathcal{A}^f \tag{5.7}$$

for all $f \in \mathfrak{F}$ for all g.

Let us now consider a stochastic algorithm whose update is given as

$$\theta_{t+1} = \mathcal{G}^f(\theta_t, \mathbf{U}_{t+1}) \ , \tag{5.8}$$

with $(\mathbf{U}_t)_{t \in \mathbb{N}_>}$ i.i.d. in $\mathbb{R}^{n\lambda}$, each $\mathbf{U}_t$ distributed as $p_\mathbf{U}$ and $\mathcal{G} : \Theta \times \mathbb{R}^{n \times \lambda} \to \Theta$ a measurable update function. We denote the function optimized f as upperscript to the update function $\mathcal{G}$. For instance, the algorithm formalized in Definition 1 can be considered, in this case the update function $\mathcal{G}$ has the structure given in (4.6).

In the generalization of the invariance definition to a stochastic algorithm, we want that not especially the same random numbers are chosen when optimizing on f or f_g. We allow a possible *coupling* of the random numbers that depends on g but also on the current state of the algorithm and that should preserve the distribution $\mathbf{U}$. More precisely we define invariance in the following manner.

Definition 5. *The algorithm defined with $\mathcal{G} : \Theta \times \mathbb{R}^{n\lambda} \to \Theta$ and the distribution $p_\mathbf{U}$ is invariant with respect to the search space transformations induced by a group action if there exists a group morphism $\Phi : (G, *) \to (S_\Theta, \circ)$ and $\psi : (G, *) \to \left(\Theta \times \mathbb{R}^{n\lambda} \to \mathbb{R}^{n\lambda}\right)$ with (i) for all g, θ, $\mathbf{u} \mapsto \psi_g(\theta, \mathbf{u})$ is bijective, (ii) $\psi_g(\theta, \mathbf{U})$ is distributed as $\mathbf{U}$ and (iii) $\psi_{g^{-1}}(\Phi_g(\theta), \psi_g(\theta, \mathbf{u})) = \mathbf{u}$ for all $\theta, g, \mathbf{u}$, such that*

$$\mathcal{G}^f(\theta, \mathbf{u}) = \Phi_g^{-1} \circ \mathcal{G}^{f_g}(\Phi_g(\theta), \psi_g(\theta, \mathbf{u})) \ .$$

for all $g \in G$, for all $\theta \in \Theta$, for all $\mathbf{u} \in \mathbb{R}^{n\lambda}$ for all $f \in \mathfrak{F}$.

Note that this type of coupling for $\mathbf{u}$ when considering invariance of stochastic systems is classical in the control context [76]. The definition 5 can be visualized by a commutative diagram as in Figure 5.2.

This definition is a generalization of the invariance definition given in [24] where we omitted the coupling for $\mathbf{u}$.

Applying the previous invariance definition for different groups, we obtain formal definitions for well-known invariances in the search space:

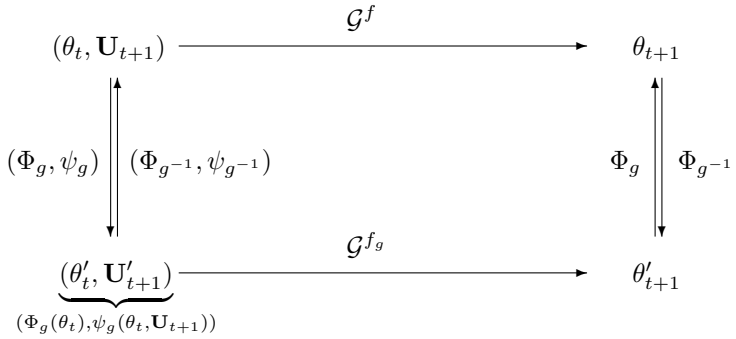

Figure 5.2: Commutative diagram for invariance with respect to transformations induced by a group G. The associated definition is given in Definition 5.

1. **translation invariance:** from the group $(\mathbb{R}^n, +)$ acting on $\mathfrak{F}$ via $\mathbf{x}_0 \in \mathbb{R}^n \mapsto (\mathbf{x} \to f(\mathbf{x} - \mathbf{x}_0))$
2. **scale invariance:** from the group $\mathbb{R}^+$ endowed with the product in $\mathbb{R}$ acting on $\mathfrak{F}$ via $\alpha \in \mathbb{R}^+ \mapsto (\mathbf{x} \to f(\alpha \mathbf{x}))$
3. **rotational invariance:** from the group special orthogonal $\mathrm{SO}(n, \mathbb{R})$ endowed with the matrix multiplication acting on $\mathfrak{F}$ via $\mathbf{R} \in \mathrm{SO}(n, \mathbb{R}) \mapsto (\mathbf{x} \to f(\mathbf{R}\mathbf{x}))$
4. **affine invariance:** from the affine group $\mathrm{Aff}(\mathbb{R}^n) = \mathbb{R}^n \rtimes \mathrm{GL(n, \mathbb{R})}$ endowed with the product multiplication given by: $(M, v) \cdot (N, w) = (MN, v + Mw)$ for $M, N \in \mathrm{GL(n, \mathbb{R})}$ and $v, w \in \mathbb{R}^n$ that is acting on $\mathfrak{F}$ via $(\mathbf{B}, \mathbf{b}) \in (\mathrm{GL(n, \mathbb{R})}, \mathbb{R}^n) \mapsto (\mathbf{x} \to f(\mathbf{B}\mathbf{x} + \mathbf{b}))$

It is straightforward to see that affine invariance implies rotational invariance, scale-invariance and translation invariance.

5.4 Affine-invariance of CMA-ES

Affine-invariance relating to invariance to affine transformations $\mathbf{x} \mapsto \mathbf{B}\mathbf{x} + \mathbf{b}$, $\mathbf{B} \in \mathrm{GL(n, \mathbb{R})}$ of the search space is a highly desirable property. Well-known affine-invariant algorithms include Newton methods [40], Nelder-Mead [70] or the quasi-Newton method BFGS [40]. In this section we prove the affine-invariance of the CMA-ES method given a small modification of the algorithm where the step-size update in (4.11) does not use the cumulative path defined in (4.10) but instead the vector $\mathbf{C}_t^{-1/2}\mathbf{p}_{t+1}$, i.e. the step-size update reads

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{C}_t^{-1/2}\mathbf{p}_{t+1}\|}{E[\|\mathcal{N}(0, I_d)\|]} - 1\right)\right) . \tag{5.9}$$

Theorem 1 (Affine Invariance of CMA-ES)**.** *The CMA-ES algorithm defined in Section 4.2.1 where the cumulative path for the step-size $\mathbf{p}_{t+1}^\sigma$ (see (4.10)) is replaced by $\mathbf{C}_t^{-1/2}\mathbf{p}_{t+1}$ is affine invariant according to Definition 5. More precisely, the homomorphism defining the state space transformation is defined as:*

$$\Phi_{(\mathbf{B},\mathbf{b})}(\mathbf{X}, \sigma, \mathbf{C}, \mathbf{p}) = \left(\mathbf{B}^{-1}(\mathbf{X} - \mathbf{b}), \sigma, \mathbf{B}^{-1}\mathbf{C}(\mathbf{B}^{-1})^T, \mathbf{B}^{-1}\mathbf{p}\right) . \tag{5.10}$$

In other words consider $\theta_t = (\mathbf{X}_t, \sigma_t, \mathbf{C}_t, \mathbf{p}_t)$ running on f and $\theta'_t = (\mathbf{X}'_t, \sigma'_t, \mathbf{C}'_t, \mathbf{p}'_t)$ running on $f(\mathbf{B}\mathbf{x} + \mathbf{b})$ with $(\mathbf{X}'_0, \sigma'_0, \mathbf{C}'_0, \mathbf{p}'_0) = \Phi_{(\mathbf{B},\mathbf{b})}(\mathbf{X}_0, \sigma_0, \mathbf{C}_0, \mathbf{p}_0)$. Then for all t the sequences θ_t and θ'_t are coupled via

$$(\mathbf{X}'_t, \sigma'_t, \mathbf{C}'_t, \mathbf{p}'_t) = \left(\mathbf{B}^{-1}(\mathbf{X}_t - \mathbf{b}), \sigma_t, \mathbf{B}^{-1}\mathbf{C}_t(\mathbf{B}^{-1})^T, \mathbf{B}^{-1}\mathbf{p}_t\right) \tag{5.11}$$

with the following coupling for the random numbers:

$$\mathbf{U}'_{t+1} = \psi_{(\mathbf{B},\mathbf{b})}(\theta_t, \mathbf{U}_{t+1}) = \left(\ldots, {\mathbf{C}'_t}^{-1/2}\mathbf{B}^{-1}\mathbf{C}_t^{1/2}\mathbf{U}^i_{t+1}, \ldots\right) \tag{5.12}$$

with the convention that the notations with prime refer to variables optimizing on the transformed function $f(\mathbf{Bx}+\mathbf{b})$.

Remark that due to the fact that Φ is a homomorphism, we directly obtain the inverse mapping to go from θ'_t to θ_t, namely

$$(\mathbf{X}_t, \sigma_t, \mathbf{C}_t, \mathbf{p}_t) = \Phi_{(\mathbf{B}^{-1}, -\mathbf{B}^{-1}\mathbf{b})}(\mathbf{X}'_t, \sigma'_t, \mathbf{C}'_t, \mathbf{p}'_t) \tag{5.13}$$

$$= (\mathbf{BX}'_t + \mathbf{b}, \sigma'_t, \mathbf{BC}'_t\mathbf{B}^T, \mathbf{Bp}_t) \tag{5.14}$$

Proof. *Given the definition of the homomorphism in* (5.10)*, we need to prove that if* (5.11) *is satisfied at time* t*, then it is satisfied at time* $t+1$*. Assume then that*

$$(\mathbf{X}'_t, \sigma'_t, \mathbf{C}'_t, \mathbf{p}'_t) = \left(\mathbf{B}^{-1}(\mathbf{X}_t - \mathbf{b}), \sigma_t, \mathbf{B}^{-1}\mathbf{C}_t(\mathbf{B}^{-1})^T, \mathbf{B}^{-1}\mathbf{p}_t\right) \ , \tag{5.15}$$

and that we run one iteration of CMA-ES on $f(\mathbf{x})$ *for advancing* θ_t *into* θ_{t+1} *and similarly we run one iteration of CMA-ES on* $f(\mathbf{Bx}+\mathbf{b})$ *(that we might denote* $\tilde{f}(x)$*) for advancing* θ'_t *into* θ'_{t+1}*. In addition for this latter step, we choose for the sampling of the multivariate normal distribution the following coupling of the random vectors* $\mathbf{U}'^i_{t+1} = {\mathbf{C}'_t}^{-1/2}\mathbf{B}^{-1}\mathbf{C}_t^{1/2}\mathbf{U}^i_{t+1}$*. Note that since the* $(\mathbf{U}^i_{t+1})_{1\leq i\leq\lambda}$ *are i.i.d. so are the* $(\mathbf{U}'^i_{t+1})_{1\leq i\leq\lambda}$*. It is not completely trivial to see that* $\mathbf{U}'^i_{t+1}$ *is still distributed as a standard multivariate normal distribution: it comes from the fact that the matrix* ${\mathbf{C}'_t}^{-1/2}\mathbf{B}^{-1}\mathbf{C}_t^{1/2}$ *is orthogonal as proven in the following equation:*

$${\mathbf{C}'_t}^{-1/2}\mathbf{B}^{-1}\mathbf{C}_t^{1/2}[{\mathbf{C}'_t}^{-1/2}\mathbf{B}^{-1}\mathbf{C}_t^{1/2}]^T = {\mathbf{C}'_t}^{-1/2}\mathbf{B}^{-1}\mathbf{C}_t^{1/2}{\mathbf{C}_t^{1/2}}^T{\mathbf{B}^{-1}}^T{\mathbf{C}'_t}^{-1/2^T} \tag{5.16}$$

$$= {\mathbf{C}'_t}^{-1/2}\mathbf{B}^{-1}\mathbf{C}_t{\mathbf{B}^{-1}}^T{\mathbf{C}'_t}^{-1/2^T} \tag{5.17}$$

$$= {\mathbf{C}'_t}^{-1/2}\mathbf{C}'_t{\mathbf{C}'_t}^{-1/2^T} \tag{5.18}$$

$$= I_d \ . \tag{5.19}$$

We note that

$${\mathbf{C}'_t}^{1/2}\mathbf{U}'^i_{t+1} = \mathbf{B}^{-1}\mathbf{C}_t^{1/2}\mathbf{U}^i_{t+1} \ . \tag{5.20}$$

This latter equation is used in the following derivation:

$$\tilde{f}(\mathbf{X}_t^{i\prime}) = \tilde{f}(\mathbf{X}'_t + \sigma'_t{\mathbf{C}'_t}^{1/2}\mathbf{U}'^i_{t+1}) \tag{5.21}$$

$$= \tilde{f}(\mathbf{B}^{-1}(\mathbf{X}_t - \mathbf{b}) + \sigma_t\mathbf{B}^{-1}\mathbf{C}_t^{1/2}\mathbf{U}^i_{t+1}) \tag{5.22}$$

$$= f(\mathbf{B}\left(\mathbf{B}^{-1}(\mathbf{X}_t - \mathbf{b}) + \sigma_t\mathbf{B}^{-1}\mathbf{C}_t^{1/2}\mathbf{U}^i_{t+1}\right) + \mathbf{b}) \tag{5.23}$$

$$= f(\mathbf{X}_t + \sigma_t\mathbf{C}_t^{1/2}\mathbf{U}^i_{t+1}) \ . \tag{5.24}$$

From the fact that $\tilde{f}(\mathbf{X}'_t + \sigma'_t{\mathbf{C}'_t}^{1/2}\mathbf{U}'^i_{t+1}) = f(\mathbf{X}_t + \sigma_t\mathbf{C}_t^{1/2}\mathbf{U}^i_{t+1})$*, we deduce that*

$$\mathbf{U}'^{i:\lambda}_{t+1} = {\mathbf{C}'_t}^{-1/2}\mathbf{B}^{-1}\mathbf{C}_t^{1/2}\mathbf{U}^{i:\lambda}_{t+1} \ . \tag{5.25}$$

We deduce from this equation the relation (5.15) *at time t+1. We start by proving that the step-sizes stay identical* $\sigma'_{t+1} = \sigma_{t+1}$*. We only need to prove that*

$$\|(\mathbf{C}'_t)^{-1/2}\mathbf{p}'_{t+1}\| = \|(\mathbf{C}_t)^{-1/2}\mathbf{p}_{t+1}\| \ .$$

Remark that $\|(\mathbf{C}'_t)^{-1/2}\mathbf{p}'_{t+1}\|^2 = \langle(\mathbf{C}'_t)^{-1/2}\mathbf{p}'_{t+1}, (\mathbf{C}'_t)^{-1/2}\mathbf{p}'_{t+1}\rangle = \langle\mathbf{p}'_{t+1}, \mathbf{C}'_t\mathbf{p}'_{t+1}\rangle$ *because* $\mathbf{C}'_t$ *is symmetric. In addition,* $\langle\mathbf{p}'_{t+1}, \mathbf{C}'_t\mathbf{p}'_{t+1}\rangle = \langle\mathbf{B}^{-1}\mathbf{p}_{t+1}, \mathbf{C}'_t\mathbf{B}^{-1}\mathbf{p}_{t+1}\rangle = \langle\mathbf{p}_{t+1}, {\mathbf{B}^{-1}}^T\mathbf{C}'_t\mathbf{B}^{-1}\mathbf{p}_{t+1}\rangle =$

$\langle \mathbf{p}_{t+1}, \mathbf{C}_t \mathbf{p}_{t+1} \rangle = \|(\mathbf{C}_t)^{-1/2} \mathbf{p}_{t+1}\|^2$. *Hence we have shown* $\|(\mathbf{C}'_t)^{-1/2} \mathbf{p}'_{t+1}\| = \|(\mathbf{C}_t)^{-1/2} \mathbf{p}_{t+1}\|$ *and thus that* $\sigma'_{t+1} = \sigma_{t+1}$. *We now derive the relation for the mean vector:*

$$\mathbf{X}'_{t+1} = \mathbf{X}'_t + \sigma'_t {\mathbf{C}'_t}^{1/2} \sum w_i \mathbf{U}'^{i:\lambda}_{t+1} \tag{5.26}$$

$$= \mathbf{B}^{-1}\mathbf{X}_t - \mathbf{B}^{-1}\mathbf{b} + \sigma_t \mathbf{B}^{-1} \mathbf{C}_t^{1/2} \sum w_i \mathbf{U}^{i:\lambda}_{t+1} \tag{5.27}$$

$$= \mathbf{B}^{-1}(\mathbf{X}_t + \sigma_t \mathbf{C}_t^{1/2} \sum w_i \mathbf{U}^{i:\lambda}_{t+1}) - \mathbf{B}^{-1}\mathbf{b} \tag{5.28}$$

$$= \mathbf{B}^{-1}(\mathbf{X}_{t+1} - \mathbf{b}) \ , \tag{5.29}$$

and for the evolution path

$$\mathbf{p}'_{t+1} = (1-c)\mathbf{p}'_t + \alpha_c {\mathbf{C}'_t}^{1/2} \sum w_i \mathbf{U}'^{i:\lambda}_{t+1} \tag{5.30}$$

$$= (1-c)\mathbf{p}'_t + \alpha_c {\mathbf{C}'_t}^{1/2} \sum w_i {\mathbf{C}'_t}^{-1/2} \mathbf{B}^{-1} \mathbf{C}_t^{1/2} \mathbf{U}^{i:\lambda}_{t+1} \tag{5.31}$$

$$= (1-c)\mathbf{B}^{-1}\mathbf{p}_t + \alpha_c \mathbf{B}^{-1} \mathbf{C}_t^{1/2} \sum w_i \mathbf{U}^{i:\lambda}_{t+1} \tag{5.32}$$

$$= \mathbf{B}^{-1}\mathbf{p}_{t+1} \ . \tag{5.33}$$

It remains to show that $\mathbf{C}'_{t+1} = \mathbf{C}_{t+1}$*:*

$$\mathbf{C}'_{t+1} = (1 - c_1 - c_\mu)\mathbf{C}'_t + c_1 \mathbf{p}'_{t+1} {\mathbf{p}'_{t+1}}^T + c_\mu \sum_{i=1}^{\mu} w_i {\mathbf{C}'_t}^{1/2} \mathbf{U}'^{i:\lambda}_{t+1} {\mathbf{U}'^{i:\lambda}_{t+1}}^T {\mathbf{C}'_t}^{1/2}$$

$$= (1 - c_1 - c_\mu)\mathbf{B}^{-1}\mathbf{C}_t\mathbf{B}^{-T} + c_1 \mathbf{B}^{-1}\mathbf{p}_{t+1}\mathbf{p}_{t+1}^T\mathbf{B}^{-T} + c_\mu \mathbf{B}^{-1} \sum w_i {\mathbf{C}_t}^{1/2} \mathbf{U}^{i:\lambda}_{t+1} {\mathbf{U}^{i:\lambda}_{t+1}}^T {\mathbf{C}_t}^{1/2} (\mathbf{B}^{-1})^T$$

$$= \mathbf{B}^{-1}\mathbf{C}_{t+1}(\mathbf{B}^{-1})^T$$

where we have used (5.25) *for the latter term and the notation* $\mathbf{B}^{-T}$ *for* $(\mathbf{B}^{-1})^T$.

Affine-invariance is certainly one key of the success of CMA-ES. One important consequence of invariance is the optimization of all functions $g \circ f_{cq}$ (where f_{cq} is a convex quadratic function and $g : f(\mathbb{R}^n) \to \mathbb{R}$ is strictly increasing) "in the same manner". Concretely on $g \circ f_{cq}$, after an adaptation phase where in particular the covariance matrix becomes proportional to the inverse of the Hessian matrix of f_{cq}, the CMA-ES algorithm optimizes the function like the sphere function $f(\mathbf{x}) = \|\mathbf{x}\|^2$, that is like the *easiest* convex-quadratic f_{cq} to optimize. So asymptotic convergence rates of CMA-ES on all $g \circ f_{cq}$ correspond to the asymptotic convergence rate on the sphere. Note that the asymptotic convergence rate does not take into account the cost of the adaptation time. However, this adaptation time can be important depending on the spectrum of the Hessian of f_{cq} and on the initial covariance matrix. The cost of the adaptation depending on the spectrum of the Hessian matrix of f_{cq} is quantified in [50].

Interestingly, it seems that using cumulative step-size adaptation (CSA) as step-size adaptation mechanism for CMA-ES requires a modification for the overall algorithm to be affine invariant according to the Definition 5. For other step-size mechanisms like two point adaptation [51], it is however clear that the overall algorithm will also be affine invariant. In addition, we conjecture that while the default CSA-CMA-ES might not satisfy Definition 5 for each iteration step, the definition should be satisfied for the invariant measure underlying the algorithm.

Remark also that affine-invariance is not the only ingredient to ensure that convex-quadratic functions are optimized after an adaptation stage like the sphere. The stability of the method is the other key ingredient. We typically observe that for too large learning rates c_1 or c_μ in (4.12) the method is not stable. Proving the stability of CMA-ES is one challenging open question that will be discussed Chapter 10.

5.5 Discussion on invariance and empirical testing

Invariance or lack of invariance has some impact on empirical testing. Indeed an invariant algorithm will exhibit similar performances on two functions from the same invariance class while a

non-invariant algorithm can exhibit performances that are drastically different. This point however assumes that the cost of forgetting the initialization or cost of the adaptation is not prominent in the overall cost to solve the objective function of the associated invariance class up to a given precision.

This latter assumption is specifically verified if we consider rotational invariance and CMA-ES where we observe in effect the same performance on separable ill-conditioned functions or on their rotated hence non-separable version (see for instance Figure 4.1). In contrast an algorithm like Particle Swarm Optimization (PSO) which is not rotational invariant exhibits performance that is very different for separable ill-conditioned or non-separable ill-conditioned problems. This effect was observed and quantified in [53] where for even a moderate condition number of 10^4, on non-separable problems, CMA-ES will be faster than PSO by a factor of 10^3 while for the same separable problem, PSO will be slightly faster than CMA. Moreover we have observed that PSO is unable to solve a standard ill-conditioned convex quadratic problem defined as a rotated version of the function $f_{\text{elli}}(\mathbf{x}) = \sum_{i=1}^{n} \alpha^{\frac{i-1}{n-1}} \mathbf{x}_i^2$ for condition numbers α larger than 10^4 in less than 10^7 function evaluations while CMA-ES solves the problem for α up to 10^{10} in less than 2×10^5 function evaluations for $n = 10, 20, 40$ (see [53, Figure 3] or also Figure 4.1).

Nonetheless, of course non rotational invariant algorithms can perform well on both a separable problem and its non-separable rotated version without performing identical on both functions. This is the case for instance for the NEWUOA algorithm [82] as seen in Figure 4.1 or more precisely in [20, 19].

We would like to conclude this discussion by stressing that invariance has been overlooked for decades in the evaluation of optimization algorithms. Consequently, some conclusions drawn about the good performance of some algorithms were solely due to the fact that several suite of test functions used to include mostly separable functions. We believe for instance that the large interest that happened around swarm algorithms like PSO is mainly due to this unfortunate bias towards separable functions into test function suites. Similarly genetic algorithms (GA) using crossover operators exploit separability and this has certainly induced some bias into the conclusions drawn about the performance of real-coded GAs.

Chapter 6

Convergence bounds - Impact on algorithm design

Contents

Comparison-based stochastic black-box algorithms as introduced in Definition 1 are typically observed to converge linearly, that is, given the estimate of the favorite solution, $\mathbf{X}_t$, at iteration t, the following equation holds

$$\lim_{t\to\infty} \frac{1}{t} \ln \frac{\|\mathbf{X}_t - \mathbf{x}^\star\|}{\|\mathbf{X}_0 - \mathbf{x}^\star\|} = -\mathrm{CR} \tag{6.1}$$

where $\mathrm{CR} > 0$ is referred to as convergence rate[1]. With our sign convention on the convergence rate: the larger CR, the faster the algorithm converges to $\mathbf{x}^\star$.

Linear convergence is particularly taking place for step-size adaptive algorithms or for algorithms like CMA-ES that combine step-size and covariance matrix adaptation mechanisms. The *step-size is the main mechanism* related to the asymptotic convergence rate in (6.1). Indeed, as we have seen, the covariance matrix adapts the shape of the underlying metric to transform an ill-conditioned problem (for instance an ill-conditioned convex-quadratic function) into a well-conditioned one (the sphere function in the case of the ill-conditioned convex-quadratic problem). The observed convergence rate of (6.1) will correspond to the convergence rate achieved by the step-size adaptation mechanism on the well-conditioned function. Without covariance matrix adaptation, the observed convergence rate on the ill-conditioned problem would be much slower.

Given this context, this chapter focuses on *step-size adaptive* algorithms where the state $\theta_t = (\mathbf{X}_t, \sigma_t) \in \mathbb{R}^n \times \mathbb{R}^+$ encodes $\mathbf{X}_t$, the favorite solution at iteration t and σ_t a scaling parameter, the step-size. We present upper bounds on the convergence rate CR assuming some specific algorithm frameworks (i.e. fixing the number of candidate solutions λ, the sampling distribution, how the update of $\mathbf{X}_t$ is done). We consider more precisely the so-called $(1+1)$-ES, $(1,\lambda)$-ES and $(\mu/\mu,\lambda)$- ES frameworks. We exhibit a specific artificial algorithm that uses a step-size proportional to the distance to the optimum whose convergence rate achieves the upper bound. We then present

[1] Eq. (6.1) can for instance hold almost surely or with an expectation, i.e. $\lim_{t\to\infty} E\left[\frac{1}{t}\ln\frac{\|\mathbf{X}_t-\mathbf{x}^\star\|}{\|\mathbf{X}_0-\mathbf{x}^\star\|}\right] = -\mathrm{CR}$ as we will see in Chapter 7.

asymptotic estimates (for the dimension n going to infinity) of the upper bounds recovering results known in the ES field as *progress rate* results usually derived under various approximations. Those results are presented in the publications [61, 17, 62].

The bounds presented are *quantitative* (i.e. can be precisely estimated) and are observed to be tight as discussed in the end of the chapter. We also explain how the bounds are useful for algorithm design and give an overview of the publications where we have use the general approach presented here to effectively design new algorithm frameworks. The related publications are [1, 12, 13, 34].

6.1 Bounds for the $(1+1)$-ES

The first framework considered is the one of the $(1+1)$-ES algorithm that was presented indirectly through the example of the $(1+1)$-ES with one-fifth success rule in Section 4.2.2. This framework is made explicit below:

$(1+1)$-ES - **Data:** $\mathbf{X}_0 \in \mathbb{R}^n$, $\sigma_0 \in \mathbb{R}_>$, $(\mathcal{N}_t)_{t\in\mathbb{N}_>}$ i.i.d. distributed as $\mathcal{N}(0, I_d)$ independent of $\mathbf{X}_0, \sigma_0$
At iteration t:

- Sample a new solution: $\mathbf{X}_{t+1}^1 = \mathbf{X}_t + \sigma_t \mathcal{N}_{t+1}$
- Evaluate the new solution and rank it w.r.t. $\mathbf{X}_t$.
- Update $\mathbf{X}_t$ by selecting the best among $\mathbf{X}_t$ and $\mathbf{X}_{t+1}^1$: $\mathbf{X}_{t+1} = \arg\min\{f(\mathbf{X}_{t+1}^1), f(\mathbf{X}_t)\}$, i.e.
$$\mathbf{X}_{t+1} = \mathbf{X}_t + \sigma_t \mathcal{N}_{t+1} 1_{\{f(\mathbf{X}_t + \sigma_t \mathcal{N}_{t+1}) \leq f(\mathbf{X}_t)\}} \tag{6.2}$$
- Adapt σ_t (this can be an oracle here giving the step-size)

For a fixed dimension n, the upper bound derived corresponds to the maximum of the following function

$$F_{(1+1)}^{(n)} : \sigma \in \mathbb{R}_{\geq} \mapsto E[\ln^-(\|e_1 + \sigma\mathcal{N}\|)] = \frac{1}{2} E\left[\ln^-\left(1 + 2\sigma[\mathcal{N}]_1 + \sigma^2\|\mathcal{N}\|^2\right)\right] \tag{6.3}$$

where $\mathcal{N}$ is a random vector following $\mathcal{N}(0, I_d)$ and $e_1 = (1, 0, \ldots, 0)$. Then $F_{(1+1)}^{(n)}$ is well defined strictly positive on $]0, +\infty[$ and continuous on $[0, +\infty]$ by setting $F_{(1+1)}^{(n)}(+\infty) := 0$ (Lemma 1 in [61]). The function is depicted in Figure 6.1 for several dimensions. We define τ its supremum

$$\tau := \sup F_{(1+1)}^{(n)}([0, +\infty]) \ , \tag{6.4}$$

it is reached and

$$\sigma_{F_{(1+1)}^{(n)}} := \min(F_{(1+1)}^{(n)})^{-1}(\tau) \tag{6.5}$$

exists (Corollary 1 in [61]). Then a step-size adaptive $(1+1)$-ES optimizing $f : \mathbb{R}^n \to \mathbb{R}$ cannot converge faster than linear with convergence rate τ, more precisely the following result holds:

Theorem 2. *[Bound for a $(1+1)$-ES [61]] Let a $(1+1)$-ES with any step-size adaptation rule $(\sigma_t)_{t\in\mathbb{N}}$ (possibly given by an oracle) optimize $f : \mathbb{R}^n \to \mathbb{R}$. Assume $\mathcal{N}_{t+1}$ is independent of σ_t and $\mathbf{X}_t$. Let $\mathbf{x}^\star$ be any vector of $\mathbb{R}^n$. Assume that $E[\ln\|\mathbf{X}_0 - \mathbf{x}^\star\|] < \infty$ and $E[\ln(1 + r\sigma_t/\|\mathbf{X}_t - \mathbf{x}^\star\|)] \in O(e^{c_t r})$ for $c_t \geq 0$. Then the convergence is at most linear with*

$$E[\ln\|\mathbf{X}_t - \mathbf{x}^\star\|] - \tau \leq E[\ln\|\mathbf{X}_{t+1} - \mathbf{x}^\star\|] \tag{6.6}$$

and

$$\inf_t \frac{1}{t} E[\ln\|\mathbf{X}_t - \mathbf{x}^\star\|/\|\mathbf{X}_0 - \mathbf{x}^\star\|] \geq -\tau \ . \tag{6.7}$$

It is then proven that this bound is reached on the sphere function $f(\mathbf{x}) = g(\|\mathbf{x} - \mathbf{x}^\star\|)$ where $g \in \mathcal{M}_{\mathbb{R}^+}$ for a specific algorithm where the step-size is proportional to the distance to the optimum. More precisely let us consider the $(1 + 1)$-ES with step-size

$$\sigma_t = \sigma \|\mathbf{X}_t - \mathbf{x}^\star\| \ , \tag{6.8}$$

for $\sigma > 0$. Note that this algorithm is "artificial" as it assumes the distance to the optimum.

Theorem 3. *[Linear convergence of the $(1 + 1)$-ES with step-size proportional to the optimum [61]] Let $\sigma > 0$ and the $(1 + 1)$-ES with step-size proportional to the optimum according to (6.8) optimizes the sphere function $f(\mathbf{x}) = g(\|\mathbf{x} - \mathbf{x}^\star\|^2)$ with $g \in \mathcal{M}_{\mathbb{R}^+}$. Then the sequence $(\mathbf{X}_t)_{t\in\mathbb{N}}$ converges linearly almost surely at the rate $F^{(n)}_{(1+1)}(\sigma)$ more precisely*

$$\lim_{t\to\infty} \frac{1}{t} \ln \frac{\|\mathbf{X}_t - \mathbf{x}^\star\|}{\|\mathbf{X}_0 - \mathbf{x}^\star\|} = -F^{(n)}_{(1+1)}(\sigma) < 0 \ \ a.s. \tag{6.9}$$

where $F^{(n)}_{(1+1)}$ is defined in (6.3).

Consequently the upper bound proven in Theorem 2 is reached for the $(1 + 1)$-ES with step-size proportional to the optimum and constant $\sigma = \sigma_{F^{(n)}_{(1+1)}}$ where $\sigma_{F^{(n)}_{(1+1)}}$ is defined in (6.5) as the smallest σ maximizing the function $F^{(n)}_{(1+1)}$.

Remark 1. *The proof of Theorem 3 is relatively simple and relies on the Law of Large Numbers (LLN) for independent random variables. Given the assumption that the step-size is proportional to the distance to the optimum, an update of the algorithm writes*

$$\|\mathbf{X}_{t+1} - \mathbf{x}^\star\| = \|\mathbf{X}_t - \mathbf{x}^\star\| \left\| \frac{\mathbf{X}_t - \mathbf{x}^\star}{\|\mathbf{X}_t - \mathbf{x}^\star\|} + \sigma \mathbf{Y}_{t+1} \right\| \tag{6.10}$$

where the vector $\mathbf{Y}_{t+1}$ equals

$$\mathbf{Y}_{t+1} = \mathcal{N}_{t+1} 1_{\{\|\mathbf{X}_t + \sigma\|\mathbf{X}_t - \mathbf{x}^\star\| \mathcal{N}_{t+1} - \mathbf{x}^\star\| \leq \|\mathbf{X}_t - \mathbf{x}^\star\|\}} = \mathcal{N}_{t+1} 1_{\{\|(\mathbf{X}_t - \mathbf{x}^\star)/\|\mathbf{X}_t - \mathbf{x}^\star\| + \sigma\mathcal{N}_{t+1}\| \leq 1\}}$$

(see (6.2)). Given the isotropy of the sphere function and of the multivariate-normal distribution, the random variables $\left\| \frac{\mathbf{X}_t - \mathbf{x}^\star}{\|\mathbf{X}_t - \mathbf{x}^\star\|} + \sigma \mathbf{Y}_{t+1} \right\|$ are independent identically distributed as $\|e_1 + \sigma\mathcal{N} 1_{\{\|e_1 + \sigma\mathcal{N}\| \leq 1\}}\|$. The linear convergence then relies on applying the LLN to obtain the limit of the following equation (obtained by taking the log in (6.10) and summing-up terms)

$$\frac{1}{t} \ln \frac{\|\mathbf{X}_t - \mathbf{x}^\star\|}{\|\mathbf{X}_0 - \mathbf{x}^\star\|} = \frac{1}{t} \sum_{k=0}^{t-1} \ln \left\| \frac{\mathbf{X}_k - \mathbf{x}^\star}{\|\mathbf{X}_k - \mathbf{x}^\star\|} + \sigma \mathbf{Y}_{k+1} \right\| \tag{6.11}$$

that thus converge towards $E[\ln \|e_1 + \sigma\mathcal{N} 1_{\{\|e_1 + \sigma\mathcal{N}\| \leq 1\}}\|] = -F^{(n)}_{(1+1)}(\sigma)$.

6.2 Bounds for the $(1, \lambda)$-ES

Similar results hold for a $(1, \lambda)$-ES step-size adaptive algorithm framework defined as

$(1, \lambda)$-ES - **Data:** $\mathbf{X}_0 \in \mathbb{R}^n$, $\sigma_0 \in \mathbb{R}_>$, $(\mathcal{N}^i_t)^{i=1,\ldots,\lambda}_{t\in\mathbb{N}_>}$ i.i.d. distributed as $\mathcal{N}(0, I_d)$; ind. of $\mathbf{X}_0, \sigma_0$
At iteration t:

- Sample λ new solutions: $\mathbf{X}^i_{t+1} = \mathbf{X}_t + \sigma_t \mathcal{N}^i_{t+1}$ where $\mathcal{N}^i(0, I_d)$ are λ i.i.d. samples
- Select the best amont the λ solutions:

$$\mathbf{X}_{t+1} = \arg\min\{f(\mathbf{X}^1_{t+1}), \ldots, f(\mathbf{X}^\lambda_{t+1})\}$$

- Adapt σ_t .

The function $F^{(n)}_{(1+1)}$ is then replaced by the function

$$F^{(n)}_{(1,\lambda)} : \sigma \in \mathbb{R}_{\geq} \mapsto E\left[-\ln\left(\min_{i=1,\ldots,\lambda} \|e_1 + \sigma\mathcal{N}^i\|\right)\right] \ , \tag{6.12}$$

where $\mathcal{N}^i$ are λ i.i.d. multivariate normal distributions. Define, $\tau_{(1,\lambda)}$ the supremum of the function $F^{(n)}_{(1,\lambda)}$, i.e.

$$\tau_{(1,\lambda)} := \sup_{\sigma} F^{(n)}_{(1,\lambda)}(\sigma) \tag{6.13}$$

and $\sigma_{F^{(n)}_{(1,\lambda)}}$ such that

$$\sigma_{F^{(n)}_{(1,\lambda)}} = \min F^{-1}(\tau_{(1,\lambda)}) \ .$$

Then similarly to Theorem 2, a $(1,\lambda)$-ES optimizing $f : \mathbb{R}^n \to \mathbb{R}$ cannot converge faster than linear with convergence rate $\tau_{(1,\lambda)}$ in the sense of (6.6) and (6.7). This bound is reached on the sphere function for the artificial algorithm using as step-size the distance to the optimum times $\sigma_{F^{(n)}_{(1,\lambda)}}$. Those results are briefly presented in [17].

Note that the function $F^{(n)}_{(1,\lambda)}$ is not always strictly positive contrary to the function $F^{(n)}_{(1+1)}$ (see Figure 6.1). Hence a $(1,\lambda)$-ES with step-size $\sigma_t = \sigma\|\mathbf{X}_t - \mathbf{x}^\star\|$ and σ such that $F^{(n)}_{(1,\lambda)}(\sigma) < 0$ will diverge linearly. In contrast, the $(1+1)$-ES cannot diverge as the framework ensures that $f(\mathbf{X}_t)$ cannot increase.

6.2.1 Extension to the framework with recombination: the $(\mu/\mu,\lambda)$-ES

Both previously presented frameworks have as new estimate of the optimum, $\mathbf{X}_{t+1}$, one of the previously evaluated solutions. However, in the CMA-ES algorithm, $\mathbf{X}_{t+1}$ is the result of the recombination of the μ best solutions out of λ, i.e.

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \sigma_t \sum_{i=1}^{\mu} w_i \mathcal{N}^{i:\lambda}_{t+1} \tag{6.14}$$

where the λ candidate solutions $\mathbf{X}_t + \sigma_t\mathcal{N}^i_{t+1}$ have been ranked, the index $i{:}\lambda$ denotes the i^{th} best solution, i.e.

$$f(\mathbf{X}_t + \sigma_t\mathcal{N}^{1:\lambda}_{t+1}) \leq \ldots \leq f(\mathbf{X}_t + \sigma_t\mathcal{N}^{\lambda:\lambda}_{t+1}) \ , \tag{6.15}$$

and $(w_i)_{1\leq i\leq\mu}$ are typically strictly positive weights summing to one, i.e., $\sum_{i=1}^{\mu} w_i = 1$. It is natural to ask the question about the generalization of the bounds for this recombination framework. Note that for $\mu = 1$ we recover the $(1,\lambda)$-ES framework. More precisely the $(\mu/\mu,\lambda)$-ES framework is defined as:

$(\mu/\mu, \lambda)$-ES - **Data:** $\mathbf{X}_0 \in \mathbb{R}^n$, $\sigma_0 \in \mathbb{R}_>$, $(\mathcal{N}_t^i)_{t\in\mathbb{N}_>}^{i=1,\ldots,\lambda}$ i.i.d. distributed as $\mathcal{N}(0, I_d)$; ind. of $\mathbf{X}_0, \sigma_0$, weights $(w_i)_{1\leq i\leq\lambda} \in \mathbb{R}^\lambda$
At iteration t:

- Sample λ new solutions: $\mathbf{X}_{t+1}^i = \mathbf{X}_t + \sigma_t \mathcal{N}_{t+1}^i$ where $\mathcal{N}^i(0, I_d)$ are λ i.i.d. samples
- Evaluate the solutions and rank the λ best solutions according to f:

$$f(\mathbf{X}_t + \sigma_t \mathcal{N}_{t+1}^{1:\lambda}) \leq \ldots \leq f(\mathbf{X}_t + \sigma_t \mathcal{N}_{t+1}^{\lambda:\lambda}) \ , \tag{6.16}$$

- Recombine the μ best solutions:

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \sigma_t \sum_{i=1}^{\mu} w_i \mathcal{N}_{t+1}^{i:\lambda} \tag{6.17}$$

- Adapt σ_t .

It turns out that in the proof of Theorem 2, the key to prove that the bound holds for any f exploits the fact that $\mathbf{X}_{t+1}$ is one of the previously evaluated solutions. Hence the proof does not generalize. However we can still define an equivalent to $F_{(1+1)}^{(n)}$ (in (6.3)) and $F_{(1,\lambda)}^{(n)}$ (in (6.12)) as

$$F_{(\mu/\mu,\lambda)}^{(n)}(\sigma) := -E\left[\ln\left\|e_1 + \sigma\sum_{i=1}^{\mu} w_i \mathcal{N}^{i:\lambda}\right\|\right] \tag{6.18}$$

where the random vectors $\mathcal{N}^{i:\lambda}$ result from the selection among λ multivariate normal distributions $\mathcal{N}^i$ according to

$$\|e_1 + \sigma\mathcal{N}^{1:\lambda}\| \leq \ldots \leq \|e_1 + \sigma\mathcal{N}^{1:\lambda}\| \ .$$

The quantity $F_{(\mu/\mu,\lambda)}^{(n)}(\sigma)$ corresponds to the (linear) convergence rate on the sphere function $g(\|\mathbf{x} - \mathbf{x}^\star\|)$ for $g \in \mathcal{M}_{\mathbb{R}^+}$ of the $(\mu/\mu, \lambda)$-ES with step-size proportional to the optimum equal at iteration t to $\sigma\|\mathbf{X}_t - \mathbf{x}^\star\|$. The function $F_{(\mu/\mu,\lambda)}^{(n)}$ has been studied in details in [62, 63]. If $\mu \leq \lambda/2$ and $n \geq 2$, then there exists an optimal step-size $\sigma_{F_{(\mu/\mu,\lambda)}^{(n)}}$ such that

$$F_{(\mu/\mu,\lambda)}^{(n)}(\sigma_{F_{(\mu/\mu,\lambda)}^{(n)}}) = \sup_{\sigma\geq 0} F_{(\mu/\mu,\lambda)}^{(n)}(\sigma) \ . \tag{6.19}$$

Then the $(\mu/\mu, \lambda)$-ES with step-size $\sigma_t = \sigma_{F_{(\mu/\mu,\lambda)}}\|\mathbf{X}_t - \mathbf{x}^\star\|$ converges linearly and its convergence rate is an upper bound on the convergence rate of all step-size adaptive $(\mu/\mu, \lambda)$-ES *on spherical functions* [62, 63].[2]

6.3 Asymptotic estimates of convergence rates - Recovering the progress rate rigorously

The functions $F_{(1+1)}^{(n)}$, $F_{(1,\lambda)}^{(n)}$ and $F_{(\mu/\mu,\lambda)}^{(n)}$ (generically denoted $F^{(n)}$ in this section) play an important role because they give bounds on the convergence rates of the $(1+1)$-ES, $(1, \lambda)$-ES and $(\mu/\mu, \lambda)$-ES algorithms respectively. Interestingly, it is possible to derive the limit for a fixed σ of $nF^{(n)}(\sigma/n)$ for n to infinity. Hence, asymptotic estimates of the convergence rate of ESs with step-size proportional to the optimum can be derived and consequently asymptotic estimates of the bounds they provide can be obtained. More precisely the following result holds:

[2] We assume here that a set of weights is fixed.

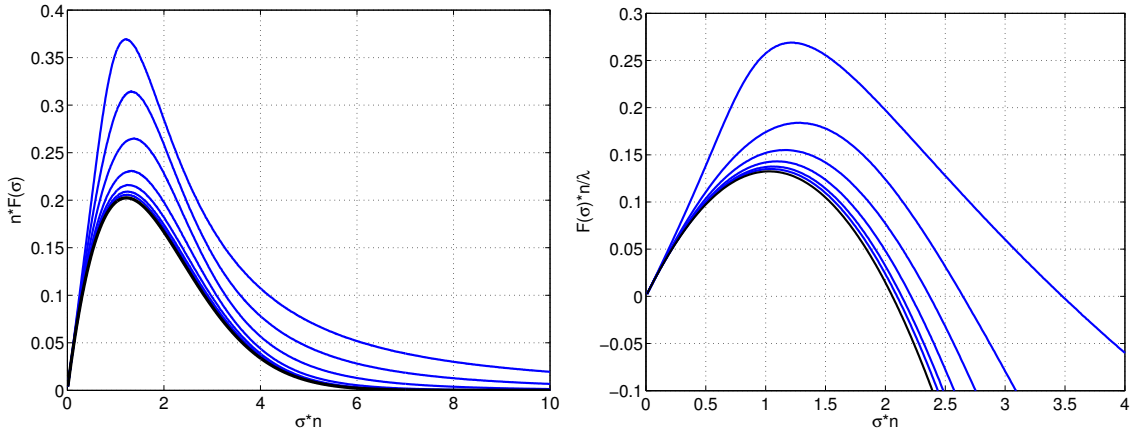

Figure 6.1: **Left**: Convergence rate of the $(1+1)$-ES with step-size proportional to the optimum given in (6.3) for dimension $n = 2, 3, 10, 20, 40, 80$ (top to bottom in blue). On the x-axis, σn and on the y-axis $nF^{(n)}_{(1+1)}(\sigma)$ are plotted. The limit of $F^{(n)}_{(1+1)}$ given in (6.20) is depicted in black. **Right:** Convergence rate of the $(1, 4)$-ES normalized by $\lambda = 4$ with step-size proportional to the optimum, i.e., on the x-axis σn, on the y-axis $nF^{(n)}_{(1,\lambda)}(\sigma)/\lambda$. Dimensions $n = 2, 3, 10, 20, 40, 80$ (top to bottom in blue). In black is depicted the limit of $nF^{(n)}_{(1,\lambda)}(\sigma/n)/\lambda$ given in (6.21). The plots of $F^{(n)}_{(1+1)}$ and $F^{(n)}_{(1,\lambda)}$ are obtained by Monte-Carlo integration with 10^7 samples for each data point.

Theorem 4. *The convergence rate* $F^{(n)}_{(1+1)}(\sigma)$ *satisfies*

$$\lim_{n\to\infty} nF^{(n)}_{(1+1)}\left(\frac{\sigma}{n}\right) = \frac{\sigma}{\sqrt{2\pi}} \exp\left(-\frac{\sigma^2}{8}\right) - \frac{\sigma^2}{2}\Phi\left(-\frac{\sigma}{2}\right) \quad . \tag{6.20}$$

where Φ *is the cumulative distribution function of standard normal distribution.*

The previous theorem tells us that the convergence rate of the $(1 + 1)$-ES with step-size proportional to the optimum expressed in (6.9) goes to zero with the dimension like $1/n$. It provides in addition a quantitative estimate of the convergence rate for n large. This linear scaling with respect to the dimension of the convergence rate is typically observed for ES algorithms like CMA-ES on spherical functions.

The proof of this theorem is sketched in [12], one assumption is however not carefully verified but just assumed, namely the uniform integrability of a sequence of random variables. The verification of this assumption was done in details for the $(\mu/\mu, \lambda)$-ES framework in [63] and follows the same line for the $(1+1)$-ES. For the sake of completeness we provide the detailed proof of Theorem 4 in the Appendix.

Similarly to Theorem 4, the following holds for the $(\mu/\mu, \lambda)$-ES framework (and thus for the $(1, \lambda)$ framework).

Theorem 5. *The convergence rate* $F^{(n)}_{(\mu/\mu,\lambda)}$ *satisfies*

$$\lim_{n\to\infty} nF^{(n)}_{(\mu/\mu,\lambda)}\left(\frac{\sigma}{n}\right) = -\left(\frac{\sigma^2}{2}\sum_{i=1}^{\mu} w_i^2 + \sigma\sum_{i=1}^{\mu} w_i E[\mathcal{N}^{i:\lambda}]\right) \tag{6.21}$$

where $\mathcal{N}^{i:\lambda}$ *is the* i^{th} *order statistic of* λ *independent standard normal distributions with mean* 0 *and variance* 1*, i.e., the* i^{th} *smallest of* λ *independent variables* $\mathcal{N}^i \sim \mathcal{N}(0, 1)$.

This theorem is proven in [13] with the detailed proof for the uniform integrability done in [63].

Similarly to the $(1 + 1)$-ES case, this theorem shows that the convergence rate of the $(\mu/\mu, \lambda)$-ES with step-size proportional to the optimum goes to zero like $1/n$.

One interesting aspect of the asymptotic formula (6.21) is that it is explicit and simple: it is a quadratic polynomial in sigma. Its maximum with respect to σ and w_i can be easily derived as

$$\frac{1}{2}\sum_{i=1}^{\mu} E(\mathcal{N}^{i:\lambda})^2$$

and the optimal weights associated equal

$$w_i^{\text{opt}} = -\frac{E(\mathcal{N}^{i:\lambda})}{\sum_{i=1}^{\mu} |E(\mathcal{N}^{i:\lambda})|} \quad . \tag{6.22}$$

Hence we see that if we impose positive weights, then optimally $\mu = \lambda/2$ with the weights given by the previous equation. If we also assume that we can have negative weights then optimally $\mu = \lambda$ with the weights given by the previous equation. This finding about optimal weights was for the first time published in [5].

Those results are not entirely new in the sense that the quantities $\lim_{n\to\infty} nF^{(n)}(\sigma/n)$ turn out to coincide with the so-called *progress rate* [83, 84, 26] derived under various approximations as an estimate of

$$nE\left[\frac{\|\mathbf{X}_t\| - \|\mathbf{X}_{t+1}\|}{\|\mathbf{X}_t\|}\middle|\mathbf{X}_t\right]$$

[83, 84, 26]. Our contributions are (i) to have connected the convergence rate of a specific algorithm (ES with step-size proportional to the optimum) to the progress rate and to bounds on convergence rates of evolution strategies and (ii) to have shown that fully rigorous mathematical analysis are amenable for ESs while progress rate results relied before on approximations that needed to be validated with simulations.

6.4 Discussion

We discuss in this section how the bounds derived, the algorithm with step-size proportional to the optimum are tightly related to algorithm design.

6.4.1 On the tightness of the bounds

We first need to argue on the tightness of the bounds obtained. The relative tightness of the bounds is observed on simulations. We can typically observe that real step-size adaptive algorithms can achieve convergence rates close to a factor of two or less to the theoretically derived bounds. This is illustrated in Figure 6.2[3]. With a mildly tuned damping parameter d_σ (see (4.11)), the CSA algorithm (i.e. CMA without covariance matrix adaptation) achieves a convergence which is less than a factor of two from the bounds for $n = 5, 10, 20, 40, 100$. With default damping, CSA is within a factor of two from the bounds for $n = 40, 100$.

6.4.2 On algorithm design

The bounds derived via the convergence rate of the artificial algorithm with step-size proportional to the optimum are used when designing algorithms in different manners:

1. Compare how close step-size adaptation mechanisms can be from the bounds and identify defects in step-size adaptation mechanisms. This point is related to the fact that the bounds are "tight". Practically, if the convergence rates on the sphere are of the order of a factor of two from the bounds, we know that there is no need to tune further the algorithm on the sphere function.

[3]This plot was produced by Asma Atamna in the context of the work published in [51]. I would like to kindly thank her for providing this plot.

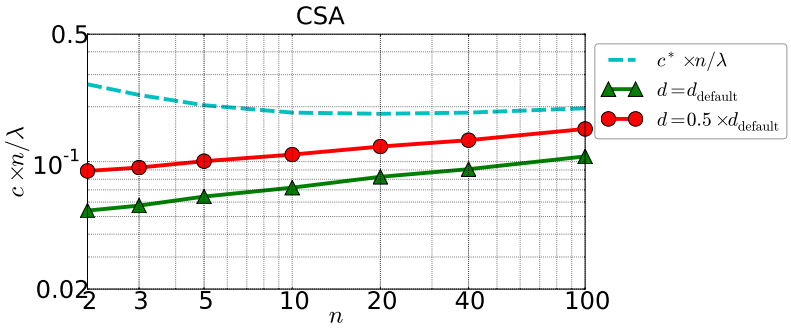

Figure 6.2: Illustration of bounds tightness. In cyan the bounds for the $(\mu/\mu, \lambda)$-ES given by the maximum of $F^{(n)}_{(\mu/\mu,\lambda)}$ normalized by λ for a given dimension n and given in (6.19) are depicted. The lambda chosen corresponds to the default value of CMA-ES. The convergence rate achieved by the CSA algorithm (i.e. CMA-ES with covariance matrix adaptation turned off) on the sphere function with default damping parameters is depicted in green. The convergence rate achieved by the CSA algorithm with a smaller damping parameter is shown in red.

2. Compare the effectiveness of different frameworks by comparing the optimal convergence rate for each framework. For instance the bound of the convergence rate of the $(1 + 1)$-ES using mirroring and sequential selection gives a 16% speedup compared to the $(1 + 1)$-ES (see next section and [12]).

3. Quantify for a given framework the influence of some parameters, or optimal parameters. For instance optimal λ and μ parameters can be identified, or optimal recombination weights can also be identified like in (6.22). For the 1/5-success rule, the parameter 1/5 as target success probability is a compromise between the asymptotic probability of success of the $(1 + 1)$-ES with step-size proportional to the optimum (and the proportionality constant achieving maximal convergence rate) on the sphere and the optimal probability of success on the so-called corridor function.

6.4.3 Designing new algorithm frameworks

We have been using the algorithm model with step-size proportional to the optimum to design new algorithm frameworks. In particular we have been studying derandomization within ESs by replacing i.i.d. samplings of multivariate normal distributions by *mirrored samples*. Roughly speaking it means given that we have a solution sampled according to $\mathbf{X}_t + \sigma_t \mathcal{N}^i(0, I_d)$, then the vector $-\mathcal{N}^i(0, I_d)$ will also be used to generate another candidate solution [34, 13, 12].

We have been deriving and simulating finite and asymptotic convergence rates for the different frameworks with step-size proportional to the optimum. We have proven that the $(1+1)$-ES with mirrored sampling and sequential selection (that concludes an iteration if an offspring is better than the current parent in which case the evaluation of some mirrored samples can be skipped), improves by 16 % the $(1 + 1)$-ES with an asymptotic convergence rate of 0.235 versus 0.202 for the $(1 + 1)$-ES [12]. We have also investigated mirrored sampling with weighted recombination, where we have added two heuristics: (i) select at most one of the two mirrored solutions and (ii) selective mirroring that consists in mirroring a certain percentage of the worst only. We have derived an asymptotic convergence rate of 0.390 while 0.25 is the best known convergence rate with positive weights, i.e. giving an improvement by 56 % [13]. Note that those improvements refer to convergence rates normalized by the number of function evaluations per iteration.

Chapter 7

Linear convergence via Markov chain stability analysis

Contents

Some natural theoretical questions, arising when studying an optimization algorithm, are whether the algorithm converges, under which conditions (on the class of functions, on the algorithm parameters, ...) and at which rate. We have ample empirical evidences of the *linear convergence* of adaptive comparison-based algorithms like CMA-ES or the $(1+1)$-ES with one-fifth success rule on wide classes of functions (see Fig 7.1 for an illustration of the linear convergence on the sphere function). However the proofs are relatively challenging, in particular compared to how easy some proofs of convergence can be achieved for stochastic algorithms like the pure random search or a $(1+1)$-ES with constant step-size. Arguably, the comparison-based property makes convergence proofs (together with convergence rates) harder has one has a weaker control on the objective function decrease.

One crucial aspect for the *linear* convergence is a proper control of the step-size, a constant step-size leading to a sub-linear convergence rate similar to the convergence rate of the pure random search (see Fig 7.1 right). We have seen in Chapter 6 that a "perfect" step-size on the sphere function would be proportional to the optimum with a proportionality constant well-chosen (as the sigma where the maximum of the functions $F_{(1+1)}$, $F^{(n)}_{(1,\lambda)}$, ... is reached). In this case, the proof of linear convergence is quite straightforward relying on the application of the Law of Large Numbers (LLN) for independent random variables to the quantity

$$\frac{1}{t}\ln\frac{\|\mathbf{X}_t-\mathbf{x}^\star\|}{\|\mathbf{X}_0-\mathbf{x}^\star\|}=\frac{1}{t}\sum_{k=0}^{t-1}\ln\frac{\|\mathbf{X}_{k+1}-\mathbf{x}^\star\|}{\|\mathbf{X}_k-\mathbf{x}^\star\|} \tag{7.1}$$

(see Remark 1). The approach we have developed which is presented in this chapter can be seen as a generalization of the proof with step-size proportional to the optimum. Using perfect

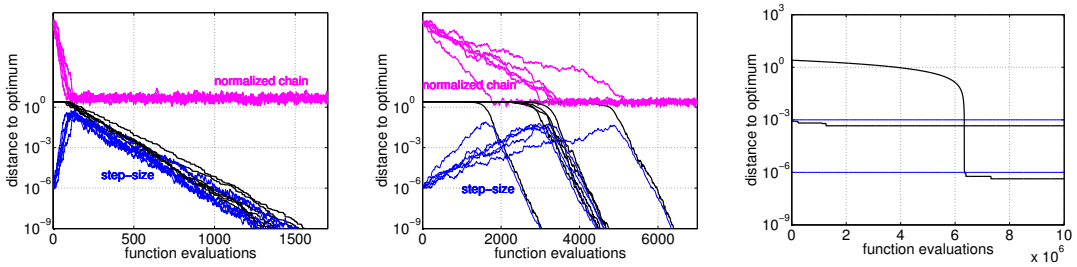

Figure 7.1: Convergence simulations on spherical functions $f(\mathbf{x}) = g(\|\mathbf{x}\|)$ for $g \in \mathcal{M}_{\mathbb{R}^+}$ in dimension $n = 10$. Left: Simulation of the $(1+1)$-ES with one-fifth success rule (see Section 4.2.2, step-size update of (4.22) implemented with $\gamma = \exp(1/3)$). Middle: xNES(see Section 4.3.4 and [24] for parameters used) using $\lambda = 4 + \lfloor 3 \ln n \rfloor$ and $\lfloor \lambda/2 \rfloor$ positive weights equals to $w_i = \ln\left(\frac{\lambda}{2} + \frac{1}{2}\right) - \ln i$ (default weights for the CMA-ES algorithm). Each plot is in log scale and depicts in black the distance to optimum, i.e. $\|\mathbf{X}_t\|$, in blue the respective step-size σ_t and in magenta the norm of the normalized chain $\|\mathbf{Z}_t\|$. The x-axis is the number of function evaluations corresponding thus to the iteration index t for the $(1+1)$-ES and to $\lambda \times t$ for xNES. For both simulations 6 independent runs are conduced starting from $\mathbf{X}_0 = (0.8, 0.8, \ldots, 0.8)$ and $\sigma_0 = 10^{-6}$. Right: Simulation of a $(1+1)$-ES with constant step-size. Two runs conducted with a constant step-size equal to 10^{-3} and 10^{-6}. The distance to the optimum is depicted in black and the step-size in blue.

step-size, the quantity $\|\mathbf{X}_t - \mathbf{x}^\star\|/\sigma_t$ is constant, while, as we will see, for a comparison-based step-size adaptive randomized search algorithm it will be the norm of a homogeneous Markov chain $\mathbf{Z}_t = (\mathbf{X}_t - \mathbf{x}^\star)/\sigma_t$. The study of the stability (irreducibility, positivity, Harris recurrence) of this Markov chain will allow to apply the LLN (for Harris-recurrent Markov chains) to the RHS of (7.1) and thus obtain a proof of the linear convergence of the associated algorithm. This will hold for comparison-based step-size adaptive randomized search algorithms that are *scale-invariant*. I already exploited this idea during my thesis to provide the first proof of linear convergence of a so-called self-adaptive ES on the sphere function [7]. However, I did not realize at that time how more general the approach was, in particular that the construction of the Markov chain $\mathbf{Z}_t$ can be done on so-called scaling-invariant functions (a particular case of such functions being the sphere function) satisfying for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $\rho > 0$

$$f(\mathbf{x}^\star + \mathbf{x}) \leq f(\mathbf{x}^\star + \mathbf{y}) \Leftrightarrow f(\mathbf{x}^\star + \rho\mathbf{x}) \leq f(\mathbf{x}^\star + \rho\mathbf{y})$$

provided the comparison-based step-size adaptive algorithm is translation and scale-invariant. The presentation of the methodology is the object of the manuscript [24], while the application of the methodology to prove the linear convergence of the $(1+1)$-ES with one-fifth success rule is presented in [18]. Remark furthermore that [7] is also an application of the methodology. We give here an overview of the main results presented in those papers.

The work presented here connects nicely comparison-based adaptive stochastic algorithms with Markov chain Monte Carlo (MCMC) methods. Indeed, the normalized chain $\mathbf{Z}_t$ can be seen as an associated MCMC algorithm for which we need to prove stability properties as done in the MCMC context.

The algorithms considered in this chapter are thus comparison-based step-size adaptive randomized search (CB-SARS) algorithms as defined in Definition 1 where the state of the algorithm is $\theta_t = (\mathbf{X}_t, \sigma_t)$ with $\mathbf{X}_t \in \mathbb{R}^n$ and $\sigma_t \in \mathbb{R}^+$. Note that it means that the full CMA-ES is not covered by this framework. We assume in addition that the update function in (4.4) takes the

form of $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2)$ such that one iteration step of the CB-SARS is given by

$$\mathbf{X}_{t+1}^i = \mathcal{S}ol((\mathbf{X}_t, \sigma_t), \mathbf{U}_{t+1}^i), i = 1, \ldots, \lambda \tag{7.2}$$

$$\mathcal{S} = \mathcal{O}rd(f(\mathbf{X}_{t+1}^1), \ldots, f(\mathbf{X}_{t+1}^\lambda)) \in \mathfrak{S}(\lambda) \tag{7.3}$$

$$\mathbf{X}_{t+1} = \mathcal{F}_1\left((\mathbf{X}_t, \sigma_t), \mathbf{U}_{t+1}^{\mathcal{S}(1)}, \ldots, \mathbf{U}_{t+1}^{\mathcal{S}(\lambda)}\right) \tag{7.4}$$

$$\sigma_{t+1} = \mathcal{F}_2\left(\sigma_t, \mathbf{U}_{t+1}^{\mathcal{S}(1)}, \ldots, \mathbf{U}_{t+1}^{\mathcal{S}(\lambda)}\right) \tag{7.5}$$

with $(\mathbf{U}_t)_{t\in\mathbb{N}_>}$ i.i.d. distributed according to $p_{\mathbf{U}}$ with each $\mathbf{U}_t$ admitting a representation as $\mathbf{U}_t = (\mathbf{U}_t^1, \ldots, \mathbf{U}_t^\lambda) \in (\mathbb{R}^n)^\lambda$. In (7.4), the $\mathcal{O}rd$ function ranks the solutions and extracts the permutation $\mathcal{S}$ containing the indexes of the ordered candidate solutions $\mathbf{X}_{t+1}^i$. A CB-SARS following the previous equations will be identified to the triplet $(\mathcal{S}ol, (\mathcal{F}_1, \mathcal{F}_2), p_{\mathbf{U}})$.

Note that for the $(1+1)$-ES with one-fifth success rule, the functions $\mathcal{F}_1$ and $\mathcal{F}_2$ were already implicitly given in (4.23) and (4.24). More precisely we have

$$\mathcal{F}_1((\mathbf{x}, \sigma), \mathbf{y}) = \mathbf{x} + \sigma\mathbf{y}_1 \tag{7.6}$$

$$\mathcal{F}_2((\mathbf{x}, \sigma), \mathbf{y}) = \sigma\left((\gamma - \gamma^{-1/4})1_{\{\mathbf{y}_1 \neq 0\}} + \gamma^{-1/4}\right) \quad . \tag{7.7}$$

Remark 2. *In [24] we admit a more general setting where $\mathbf{U}_t \in \mathbb{U}^\lambda = (\mathbb{U} \times \ldots \times \mathbb{U})$ with $\mathbb{U} \subset \mathbb{R}^m$ where m is not necessarily equal to n. This more general setting allows in particular to include so-called self-adaptive algorithms where $\mathbb{U} = \mathbb{R}^{n+1}$. For the sake of clarity, we consider $\mathbb{U} = \mathbb{R}^n$ in this document.*

7.1 Construction of the homogeneous Markov chain: consequence of scale and translation invariance

We explain how the construction of the Markov chain $\mathbf{Z}_t = (\mathbf{X}_t - \mathbf{x}^\star)/\sigma_t$ associated to a CB-SARS on scaling-invariant functions derives from scale and translation invariance of the algorithm.

7.1.1 The class of scaling-invariant functions

The construction of the Markov chain candidate to be stable works on the class of scaling-invariant functions that satisfy the following definition.

Definition 6. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is scaling-invariant with respect to $\mathbf{x}^\star \in \mathbb{R}^n$, if for all $\rho > 0$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$*

$$f(\mathbf{x}^\star + \mathbf{x}) \leq f(\mathbf{x}^\star + \mathbf{y}) \Leftrightarrow f(\mathbf{x}^\star + \rho\mathbf{x}) \leq f(\mathbf{x}^\star + \rho\mathbf{y}) \quad . \tag{7.8}$$

Examples of scaling-invariant functions include $f(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^\star\|$ for any arbitrary norm on $\mathbb{R}^n$ since in this case $f(\mathbf{x}^\star + \rho\mathbf{x}) = \|\rho\mathbf{x}\| = \rho\|\mathbf{x}\| = \rho f(\mathbf{x} + \mathbf{x}^\star)$ (for $\rho > 0$). It also includes functions with non-convex sublevel sets, i.e. non-quasi-convex functions (see Fig 7.2 for an illustration).

A scaling-invariant function cannot admit any strict local optima besides $\mathbf{x}^\star$. In addition, on a line crossing $\mathbf{x}^\star$ a scaling invariant function is either constant equal to $f(\mathbf{x}^\star)$ or cannot admit a local plateau (see Proposition 3.2 in [24]). A specific class of scaling-invariant functions are positively homogeneous functions.

Definition 7 (Positively homogeneous functions)**.** *A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is said positively homogeneous with degree α if for all $\rho > 0$ and for all $\mathbf{x} \in \mathbb{R}^n$, $f(\rho\mathbf{x}) = \rho^\alpha f(\mathbf{x})$.*

From this definition follows that if a function $\hat{f}$ is positively homogeneous with degree α then $\hat{f}(\mathbf{x} - \mathbf{x}^\star)$ is scaling-invariant with respect to $\mathbf{x}^\star$ for any $\mathbf{x}^\star \in \mathbb{R}^n$. Examples of positively homogeneous functions are linear functions that are positively homogeneous functions with degree 1. Also, every function deriving from a norm is positively homogeneous with degree 1.

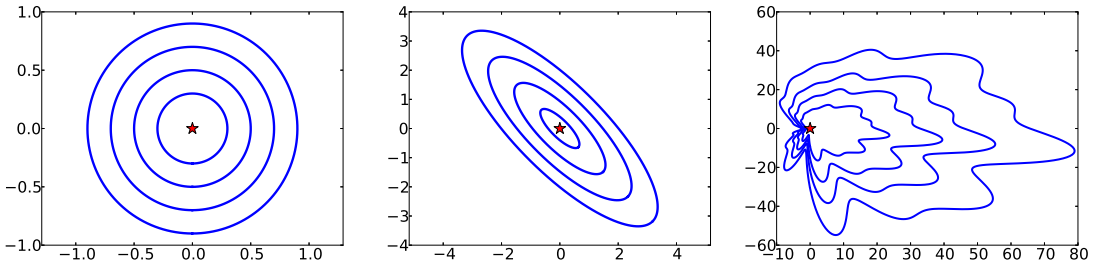

Figure 7.2: Illustration of scaling-invariant functions w.r.t. the point $\mathbf{x}^\star$ depicted with a star. The three functions are composite of $g \in \mathcal{M}$ by $f(\mathbf{x}-\mathbf{x}^\star)$ where f is a positively homogeneous function (see Definition 7). Left: composite of $g \in \mathcal{M}$ and $f(\mathbf{x}) = \|\mathbf{x}-\mathbf{x}^\star\|$. Middle: composite of $g \in \mathcal{M}$ and $f(\mathbf{x}) = (\mathbf{x}-\mathbf{x}^\star)^T A(\mathbf{x}-\mathbf{x}^\star)$ for A symmetric positive definite. Right: randomly generated scaling-invariant function from a "smoothly" randomly perturbed sphere function. Both functions on the left have convex sublevel sets contrary to the one on the right.

7.1.2 Scale and translation invariant CB-SARS

The construction of the Markov chain is a consequence of scale and translation invariance that we assume thus for the comparison-based step-size adaptive randomized search considered. More precisely we will assume that (i) the $\mathcal{S}ol$ function satisfies: for all $\alpha > 0$, for all $\mathbf{u} \in \mathbb{R}^n, (\mathbf{x}, \sigma) \in \mathbb{R}^n \times \mathbb{R}^+_>$

$$\mathcal{S}ol((\mathbf{x}, \sigma), \mathbf{u}) = \alpha \mathcal{S}ol\left(\left(\frac{\mathbf{x}}{\alpha}, \frac{\sigma}{\alpha}\right), \mathbf{u}\right) \tag{7.9}$$

(ii) the $\mathcal{F}_1$ function satisfies for all $\alpha > 0$, for all $\mathbf{y} \in \mathbb{R}^{n\lambda}, (\mathbf{x}, \sigma) \in \mathbb{R}^n \times \mathbb{R}^+_>$

$$\mathcal{F}_1((\mathbf{x}, \sigma), \mathbf{y}) = \alpha \mathcal{F}_1\left(\left(\frac{\mathbf{x}}{\alpha}, \frac{\sigma}{\alpha}\right), \mathbf{y}\right) \tag{7.10}$$

and (iii) the $\mathcal{F}_2$ function satisfies for all $\alpha > 0$, for all $\mathbf{y} \in \mathbb{R}^{n\lambda}, \sigma \in \mathbb{R}^+_>$

$$\mathcal{F}_2(\sigma, \mathbf{y}) = \alpha \mathcal{F}_2\left(\frac{\sigma}{\alpha}, \mathbf{y}\right) \quad . \tag{7.11}$$

Those three conditions ensure the scale-invariance of the underlying algorithm and the associated homomorphism of Definition 5 is $\Phi : \alpha \in \mathbb{R}^+_> \mapsto \Phi(\alpha)$ where for all $(\mathbf{x}, \sigma) \in \mathbb{R}^n \times \mathbb{R}^+_>$,

$$\Phi(\alpha)(\mathbf{x}, \sigma) = (\mathbf{x}/\alpha, \sigma/\alpha) \tag{7.12}$$

(see Proposition 2.9 in [24]). Note that related to Definition 5, no coupling for the random vectors $\mathbf{U}^i_{t+1}$ is needed such that $\psi_g(\theta, \mathbf{u}) = \mathbf{u}$. For translation invariance we more precisely assume (i) that for all $\mathbf{x}, \mathbf{x}_0 \in \mathbb{R}^n$ for all $\sigma > 0$ and for all $\mathbf{u} \in \mathbb{R}^n$

$$\mathcal{S}ol((\mathbf{x} + \mathbf{x}_0, \sigma), \mathbf{u}) = \mathcal{S}ol((\mathbf{x}, \sigma), \mathbf{u}) + \mathbf{x}_0 \tag{7.13}$$

and that (ii) for all $\mathbf{x}, \mathbf{x}_0 \in \mathbb{R}^n$ for all $\sigma > 0$ and for all $\mathbf{y} \in \mathbb{R}^{n\lambda}$

$$\mathcal{F}_1((\mathbf{x} + \mathbf{x}_0, \sigma), \mathbf{y}) = \mathcal{F}_1((\mathbf{x}, \sigma), \mathbf{y}) + \mathbf{x}_0 \quad . \tag{7.14}$$

Hence, we know then with Proposition 2.7 in [24] that the underlying comparison-based step-size adaptive randomized search is translation invariant with the associated group homomorphism Φ defined as

$$\Phi(\mathbf{x}_0)(\mathbf{x}, \sigma) = (\mathbf{x} + \mathbf{x}_0, \sigma) \text{ for all } \mathbf{x}_0, \mathbf{x}, \sigma \quad . \tag{7.15}$$

Here also no coupling for the random numbers is needed.

7.1.3 Construction of a homogeneous Markov chain

The following proposition proves that on scaling-invariant functions for a CB-SARS satisfying the scale and translation invariance conditions of the previous section, the sequence $\mathbf{Z}_t = (\mathbf{X}_t - \mathbf{x}^\star)/\sigma_t$ is a homogeneous Markov chain.

Proposition 1 ([24, Proposition 4.1]). *Consider a scaling-invariant objective function f optimized by $(\mathcal{S}ol, (\mathcal{F}_1, \mathcal{F}_2), p_{\mathbf{U}})$, a CB-SARS algorithm assumed to be translation-invariant and scale-invariant satisfying (7.9), (7.10) and (7.11). Let $(\mathbf{X}_t, \sigma_t)_{t\in\mathbb{N}}$ be the Markov chain associated to this CB-SARS. Let $\mathbf{Z}_t = \frac{\mathbf{X}_t - \mathbf{x}^\star}{\sigma_t}$ for all $t \in \mathbb{N}$. Then $(\mathbf{Z}_t)_{t\in\mathbb{N}}$ is a homogeneous Markov chain that can be defined independently of $(\mathbf{X}_t, \sigma_t)$, provided $\mathbf{Z}_0 = (\mathbf{X}_0 - \mathbf{x}^\star)/\sigma_0$ via*

$$\mathbf{Z}_{t+1}^i = \mathcal{S}ol((\mathbf{Z}_t, 1), \mathbf{U}_{t+1}^i), i = 1, \ldots, \lambda \tag{7.16}$$

$$\mathcal{S} = \mathcal{O}rd(f(\mathbf{Z}_{t+1}^1 + \mathbf{x}^\star), \ldots, f(\mathbf{Z}_{t+1}^\lambda + \mathbf{x}^\star)) \tag{7.17}$$

$$\mathbf{Z}_{t+1} = G(\mathbf{Z}_t, \mathcal{S} * \mathbf{U}_{t+1}) = G(\mathbf{Z}_t, (\mathbf{U}_{t+1}^{\mathcal{S}(1)}, \ldots, \mathbf{U}_{t+1}^{\mathcal{S}(\lambda)})) \tag{7.18}$$

where $(\mathbf{U}_t)_{t\in\mathbb{N}_>}$ is an i.i.d. sequence of random vectors distributed according to $p_{\mathbf{U}}$ and where the function G equals for all $\mathbf{z} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^{n\lambda}$

$$G(\mathbf{z}, \mathbf{y}) = \frac{\mathcal{F}_1((\mathbf{z}, 1), \mathbf{y})}{\mathcal{F}_2(1, \mathbf{y})} \ . \tag{7.19}$$

Note that because we have assumed scale-invariance and in particular (7.11) for the function $\mathcal{F}_2$, the step-size update has a specific shape. It satisfies namely

$$\sigma_{t+1} = \sigma_t \mathcal{F}_2(1, \mathbf{Y}_t) \tag{7.20}$$

where $\mathbf{Y}_t = \mathcal{S} * \mathbf{U}_{t+1} = (\mathbf{U}_{t+1}^{\mathcal{S}(1)}, \ldots, \mathbf{U}_{t+1}^{\mathcal{S}(\lambda)})$. Denoting the multiplicative step-size update as $\eta^\star$, i.e.

$$\eta^\star(\mathbf{Y}_t) = \mathcal{F}_2(1, \mathbf{Y}_t) \ , \tag{7.21}$$

the update of $\mathbf{Z}_t$ given in (7.19), reads that $\mathbf{Z}_{t+1}$ equals the mean update for step-size equal 1 divided by the step-size change $\eta^\star$.

Remark 3. *The proof of the previous proposition provided in [24] reveals that the ranking permutation $\mathcal{S}$ is the same when ranking solutions sampled from $(\mathbf{X}_t, \sigma_t)$ on $\mathbf{x} \mapsto f(\mathbf{x})$ or when ranking solutions sampled from $(\mathbf{Z}_t, 1)$ on $\mathbf{x} \mapsto f(\mathbf{x}+\mathbf{x}^\star)$ such that $\eta^\star(\mathcal{S}^f_{(\mathbf{X}_t, \sigma_t)} * \mathbf{U}_{t+1}) = \eta^\star(\mathcal{S}^{f^\star_{\mathbf{x}}}_{(\mathbf{Z}_t, 1)} * \mathbf{U}_{t+1})$.*

7.2 Sufficient Conditions for Linear Convergence

The link between stability of $\mathbf{Z}_t$ and linear convergence of the CB-SARS can now be explained. The linear convergence results from investigating (7.1) that sums the log-progresses $\ln \|\mathbf{X}_{t+1} - \mathbf{x}^\star\|/\|\mathbf{X}_t - \mathbf{x}^\star\|$. The chains $(\mathbf{X}_t, \sigma_t)_{t\in\mathbb{N}}$ and $(\mathbf{Z}_t)_{t\in\mathbb{N}}$ being connected by the relation $\mathbf{Z}_t = (\mathbf{X}_t - \mathbf{x}^\star)/\sigma_t$, the log-progress can be expressed as

$$\ln \frac{\|\mathbf{X}_{t+1} - \mathbf{x}^\star\|}{\|\mathbf{X}_t - \mathbf{x}^\star\|} = \ln \frac{\|\mathbf{Z}_{t+1}\| \eta^\star(\mathbf{Y}(\mathbf{Z}_t, \mathbf{U}_{t+1}))}{\|\mathbf{Z}_t\|} \tag{7.22}$$

where the ordered vector $\mathcal{S}^{f_{\mathbf{x}^\star}}_{(\mathbf{Z}_t, 1)} * \mathbf{U}_{t+1}$ is denoted $\mathbf{Y}(\mathbf{Z}_t, \mathbf{U}_{t+1})$ to signify its dependency in $\mathbf{Z}_t$ and $\mathbf{U}_{t+1}$, i.e.

$$\mathbf{Y}(\mathbf{z}, \mathbf{u}) = \mathcal{S}^{f_{\mathbf{x}^\star}}_{(\mathbf{z}, 1)} * \mathbf{u} = \mathcal{O}rd(\{f(\mathcal{S}ol((\mathbf{z}, 1), \mathbf{u}^i) + \mathbf{x}^\star)\}_{i=1,\ldots,\lambda}) * \mathbf{u} \ . \tag{7.23}$$

In (7.22) we use the fact that the step-size change starting from $(\mathbf{X}_t, \sigma_t)$ equals the step-size change starting from $(\mathbf{Z}_t, 1)$ (see Remark 3). Using the property of the logarithm, we obtain

$$\frac{1}{t} \ln \frac{\|\mathbf{X}_t - \mathbf{x}^\star\|}{\|\mathbf{X}_0 - \mathbf{x}^\star\|} = \frac{1}{t} \sum_{k=0}^{t-1} \ln \frac{\|\mathbf{X}_{k+1} - \mathbf{x}^\star\|}{\|\mathbf{X}_k - \mathbf{x}^\star\|} = \frac{1}{t} \sum_{k=0}^{t-1} \ln \left(\frac{\|\mathbf{Z}_{k+1}\|}{\|\mathbf{Z}_k\|} \eta^\star(\mathbf{Y}(\mathbf{Z}_t, \mathbf{U}_{t+1})) \right) \ . \tag{7.24}$$

We understand now that we can conclude to the linear convergence of the CB-SARS if the chain $\mathbf{Z}_t$ satisfies conditions that ensure that we can apply the LLN to the RHS of the previous equation. Typical conditions for a Markov chain $\mathbf{Z}_t$ to satisfy a LLN are positivity (i.e. existence of an invariant probability measure) and Harris recurrence. In this case one can conclude that

$$\frac{1}{t}\ln\frac{\|\mathbf{X}_t-\mathbf{x}^\star\|}{\|\mathbf{X}_0-\mathbf{x}^\star\|}\xrightarrow[t\to\infty]{a.s.}\int E[\ln(\eta^\star(\mathbf{Y}(\mathbf{z},\mathbf{U})))]\pi(d\mathbf{z})$$

where π is the invariant probability measure of $\mathbf{Z}_t$ and the expectation in the integral is w.r.t. $\mathbf{U}\sim p_{\mathbf{U}}$ (see [24, Theorem 5.1]). The previous convergence equation can also hold with an expectation (see [24, Theorem 5.2]).

7.3 Studying the stability of the normalized homogeneous Markov chain

We have seen that the study of the Markov chain $\mathbf{Z}_t$—in particular the establishment of *stability* properties that allow to state a strong Law of Large Numbers, that is $\frac{1}{t}\sum_{k=0}^{t-1}g(\mathbf{Z}_k)$ converges to $\pi(g)$ where π is the invariant measure of $\mathbf{Z}_t$—can allow to conclude to the linear convergence of the CB-SARS associated to the Markov chain $\mathbf{Z}_t$.

Sufficient stability conditions for proving a LLN for Markov chains are φ-irreducibility, Harris recurrence and positivity whose definitions are briefly reviewed [78].

Let $\mathbf{Z}=(\mathbf{Z}_t)_{t\in\mathbb{N}}$ be a Markov chain defined on a state space $\mathcal{Z}$ equipped with the Borel sigma-algebra $\mathcal{B}(\mathcal{Z})$. We denote $P^t(\mathbf{z},A)$, $t\in\mathbb{N}$, $\mathbf{z}\in\mathcal{Z}$ and $A\in\mathcal{B}(\mathcal{Z})$ the transition probabilities of the chain

$$P^t(\mathbf{z},A)=P_{\mathbf{z}}(\mathbf{Z}_t\in A) \tag{7.25}$$

where $P_{\mathbf{z}}$ and $E_{\mathbf{z}}$ denote the probability law and expectation of the chain under the initial condition $\mathbf{Z}_0=\mathbf{z}$. If a probability μ on $(\mathcal{Z},\mathcal{B}(\mathcal{Z}))$ is the initial distribution of the chain, the corresponding quantities are denoted P_μ and E_μ. For $t=1$, the transition probability in (7.25) is denoted $P(\mathbf{z},A)$. The chain $\mathbf{Z}$ is φ-irreducible if there exists a non-zero measure φ such that for all $A\in\mathcal{B}(\mathcal{Z})$ with $\varphi(A)>0$, for all $\mathbf{z}_0\in\mathcal{Z}$, the chain started at $\mathbf{z}_0$ has a positive probability to hit A, that is there exists $t\in\mathbb{N}_>$ such that $P^t(\mathbf{z}_0,A)>0$. A σ-finite measure π on $\mathcal{B}(\mathcal{Z})$ is said invariant if it satisfies

$$\pi(A)=\int\pi(d\mathbf{z})P(\mathbf{z},A),\ \ A\in\mathcal{B}(\mathcal{Z})\ .$$

If the chain $\mathbf{Z}$ is φ-irreducible and admits an invariant probability measure then it is called *positive*. A small set is a set C such that for some $\delta>0$ and $t>0$ and some non trivial probability measure ν_t,

$$P^t(\mathbf{z},.)\geq\delta\nu_t(.),\mathbf{z}\in C\ .$$

The set C is then called a ν_t-small set. Consider a small set C satisfying the previous equation with $\nu_t(C)>0$ and denote $\nu_t=\nu$. The chain is called aperiodic if the greatest common divisor of the set

$$E_C=\{k\geq 1: C \text{ is a } \nu_k\text{-small set with } \nu_k=\alpha_k\nu \text{ for some } \alpha_k>0\}$$

is one for some (and then for every) small set C.

A φ-irreducible Markov chain is *Harris-recurrent* if for all $A\subset\mathcal{Z}$ with $\varphi(A)>0$, and for all $\mathbf{z}\in\mathcal{Z}$, the chain will eventually reach A with probability 1 starting from $\mathbf{z}$, formally if $P_{\mathbf{z}}(\eta_A=\infty)=1$ where η_A is the *occupation time* of A, i.e. $\eta_A=\sum_{t=1}^{\infty}1_{\mathbf{Z}_t\in A}$. A (Harris-)recurrent chain admits a unique (up to a constant multiple) invariant measure [78, Theorem 10.0.4].

Typical sufficient conditions for a Law of Large Numbers to hold are φ-irreducibility, positivity and Harris-recurrence as formally stated in the next theorem:

Theorem 6. *[[78] Theorem 17.0.1] Assume that $\mathbf{Z}$ is a positive Harris-recurrent chain with invariant probability π. Then the LLN holds for any g with $\pi(|g|)=\int|g(\mathbf{x})|\pi(d\mathbf{x})<\infty$, that is for any initial state $\mathbf{Z}_0$, $\lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}g(\mathbf{Z}_k)=\pi(g)$ a.s.*

We typically derive Harris-recurrence of $(\mathbf{Z}_t)_{t\in\mathbb{N}}$ by proving Foster-Lyapunov drift conditions. We actually derive drift conditions to prove a stronger property, namely geometric ergodicity which implies then positivity and Harris-recurrence.

Geometric ergodicity translates the fact that convergence to the invariant measure takes place at a geometric rate. Different notions of geometric ergodicity do exist (see [78]) and we consider the form that appears in the following theorem. For any V, PV is defined as $PV(\mathbf{z}) := \int P(\mathbf{z}, d\mathbf{y})V(\mathbf{y})$. For a function $V \geq 1$, the V-norm for a signed measure ν is defined as

$$\|\nu\|_V = \sup_{k:|k|\leq V} |\nu(k)| = \sup_{k:|k|\leq V} |\int k(\mathbf{y})\nu(d\mathbf{y})| \ .$$

Theorem 7. *(Geometric Ergodic Theorem [78, Theorem 15.0.1]) Suppose that the chain* $\mathbf{Z}$ *is* ψ*-irreducible and aperiodic. Then the following three conditions are equivalent: (i) The chain* $\mathbf{Z}$ *is positive recurrent with invariant probability measure* π*, and there exists some petite set* $C \in \mathcal{B}^+(\mathcal{Z})$ *(such that* $\varphi(C) > 0$*),* $\rho_C < 1$*,* $M_C < \infty$*, and* $P^\infty(C) > 0$ *such that for all* $\mathbf{z} \in C$

$$|P^t(\mathbf{z}, C) - P^\infty(C)| \leq M_C \rho_C^t.$$

(ii) There exists some petite set C *and* $\kappa > 1$ *such that*

$$\sup_{\mathbf{z}\in C} E_{\mathbf{z}}[\kappa^{\tau_C}] < \infty \ .$$

(iii) There exists a petite set $C \in \mathcal{B}(\mathcal{Z})$*, constants* $b < \infty$*,* $\vartheta < 1$*, and a function* $V \geq 1$ *finite at some one* $\mathbf{z}_0 \in \mathcal{Z}$ *satisfying*

$$PV(\mathbf{z}) \leq \vartheta V(\mathbf{z}) + b1_C(\mathbf{z}), \mathbf{z} \in \mathcal{Z}. \tag{7.26}$$

Any of these three conditions imply that the following two statements hold. The set $S_V = \{\mathbf{z} : V(\mathbf{z}) < \infty\}$ *is absorbing and full, where* V *is any solution to* (7.26)*. Furthermore, there exist constants* $r > 1$ *and* $R < \infty$ *such that for any* $\mathbf{z} \in S_V$

$$\sum_t r^t \|P^t(\mathbf{z}, .) - \pi\|_V \leq RV(\mathbf{z}) \ . \tag{7.27}$$

The drift operator is defined as $\Delta V(\mathbf{z}) = PV(\mathbf{z}) - V(\mathbf{z})$. The inequality (7.26) is called a drift condition that can be re-written as

$$\Delta V(\mathbf{z}) \leq \underbrace{(\vartheta - 1)}_{<0} V(\mathbf{z}) + b1_C(\mathbf{z}) \ .$$

P is then said to admit a drift towards the set C. The previous theorem is using the notion of petite sets but small sets are actually also petite sets (see Section 5.5.2 [78]).

7.3.1 Linear Convergence of the $(1 + 1)$-ES with generalized one-fifth success rule

Using the previously sketched methodology, we have been proving the convergence of the *generalized* $(1+1)$*-ES with one-fifth success rule*, a slightly generalized version of the algorithm presented in Section 4.2.2 [18]. The algorithm first samples a candidate solution from a multivariate normal distribution centered in $\mathbf{X}_t$ and with covariance matrix $\sigma_t^2 I_d$ (where t is the iteration index):

$$\mathbf{X}_{t+1}^1 = \mathbf{X}_t + \sigma_t \mathbf{U}_{t+1}^1 \tag{7.28}$$

where $\mathbf{U}_{t+1}^1$ follows a standard multivariate normal distribution, i.e., $\mathbf{U}_{t+1}^1 \sim \mathcal{N}(0, I_d)$. The new solution is accepted and replaces $\mathbf{X}_t$ if its f-value is better than $f(\mathbf{X}_t)$, i.e.

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \sigma_t \mathbf{U}_{t+1}^1 1_{\{f(\mathbf{X}_{t+1}^1)\leq f(\mathbf{X}_t)\}} \ . \tag{7.29}$$

The *step-size* σ_t is increased in case of success and decreased otherwise [87, 41, 83]. We denote $\gamma > 1$ the increasing factor and introduce a parameter $q \in \mathbb{R}^+_>$ such that the factor for decrease equals $\gamma^{-1/q}$. Overall the step-size update reads

$$\sigma_{t+1} = \sigma_t \gamma 1_{\{f(\mathbf{x}^1_{t+1}) \leq f(\mathbf{x}_t)\}} + \sigma_t \gamma^{-1/q} 1_{\{f(\mathbf{x}^1_{t+1}) > f(\mathbf{x}_t)\}} \ . \tag{7.30}$$

When $-1/q = -1/4$, this update implements the idea to maintain a probability of success around $1/5$ and correspond to the original idea proposed by Schumer and Steiglitz [87], Devroye [41] and Rechenberg [83] described in Section 4.2.2. It is relatively straightforward to show that the generalized $(1+1)$-ES with one-fifth success rule satisfies conditions (7.9), (7.10) and (7.11) such that, according to Proposition 1, on scaling-invariant functions $\mathbf{Z}_t = \frac{\mathbf{X}_t - \mathbf{x}^\star}{\sigma_t}$ is a homogeneous Markov chain.

We investigate the stability of the Markov chain $\mathbf{Z}_t$ on a particular case of scaling-invariant functions, namely positively homogeneous functions defined in Definition 7 where we make moreover the following assumptions.

Assumption 1. *The function $f : \mathbb{R}^n \to \mathbb{R}$ is homogeneous with degree α and $f(\mathbf{x}) > 0$ for all $\mathbf{x} \neq 0$.*

This assumption implies that the function f has a *unique optimum* located w.l.o.g. in 0 (if the optimum $\mathbf{x}^\star$ is not in 0, consider $\tilde{f} = f(\mathbf{x} - \mathbf{x}^\star)$). Note that with this assumption, we exclude linear functions.

Under this hypothesis and assuming that f is continuous on $\mathbb{R}^n \backslash \{0\}$ we can establish easily the φ-irreducibility of the chain $(\mathbf{Z}_t)_{t \in \mathbb{N}}$ with respect to the Lebesgue measure provided that $\gamma > 1$ (see [18, Proposition 3.2]). Under the same assumptions we prove that the sets $D_{[l_1, l_2]}$ with $0 < l_1 < l_2$ defined as

$$D_{[l_1, l_2]} := \{\mathbf{z} \in \mathcal{Z}, l_1 \leq f(\mathbf{z}) \leq l_2\} \ . \tag{7.31}$$

are small sets [18, Lemma 3.3] and that the chain $\mathbf{Z}_t$ is aperiodic [18, Proposition 3.4]. We finally establish a drift for geometric ergodicity under the following assumption.

Assumption 2. *The function $f : \mathbb{R}^n \to [0, +\infty[$ is a positively homogeneous function with degree α and $f(\mathbf{x}) > 0$ for all $\mathbf{x} \neq 0$.*
The function f is continuously differentiable and $\alpha \leq n$. There exists $k \in \mathbb{N}_>$ and $c_0, \ldots, c_k$ in $\mathbb{R}$ such that for all $\tilde{\mathbf{z}} \in \mathcal{L}_1$ and $\mathbf{y} \in \mathbb{R}^n$, $c_{\tilde{\mathbf{z}}}, c_{\mathbf{y}} \in [0, 1]$

$$\|\nabla f(\tilde{\mathbf{z}} + c_{\tilde{\mathbf{z}}} c_{\mathbf{y}} \mathbf{y})\|^2 \leq c_0 + \sum_{i=1}^{k} c_i \|\mathbf{y}\|^i \ . \tag{7.32}$$

Theorem 8 ([18, Theorem 3.11])**.** *Consider $(\mathbf{X}_t, \sigma_t)_{t \in \mathbb{N}}$ associated to the (1+1)-ES with generalized one-fifth success rule algorithm as defined in (7.28), (7.29) and (7.30) optimizing $h = g \circ f$ where $g \in \mathcal{M}$ and $f : \mathbb{R}^n \to [0, +\infty[$ satisfies Assumption 2. Let $\mathbf{Z} = (\mathbf{Z}_t = \mathbf{X}_t / \sigma_t)_{t \in \mathbb{N}}$ be the Markov chain associated to the (1+1)-ES optimizing h defined in Proposition 1. Then the function*

$$V(\mathbf{z}) = f(\mathbf{z}) 1_{\{f(\mathbf{z}) \geq 1\}} + \frac{1}{f(\mathbf{z})} 1_{\{f(\mathbf{z}) < 1\}} \tag{7.33}$$

satisfies a drift condition for geometric ergodicity (in the sense of (7.26)) for the Markov chain $\mathbf{Z}$ if $\gamma > 1$ and

$$\frac{1}{2}\left(\frac{1}{\gamma^\alpha} + \gamma^{\alpha/q}\right) < 1 \ . \tag{7.34}$$

Interestingly, the condition (7.34) under which the drift for geometric ergodicity holds means that the expectation of the inverse of the step-size change on linear functions is smaller than one:

$$E[1/(\eta^\star)^\alpha_{\text{linear}}] < 1 \ , \tag{7.35}$$

which translates to a step-size increase on a linear function. This condition is similar to the one found to prove geometric ergodicity for the $(1, \lambda)$-ES with self-adaptation on the sphere function [7].

Remark that we have some latitude to model a given function $\tilde{f}$ as $h \circ f$ with f positively homogeneous with degree α and $h \in \mathcal{M}$. Indeed by playing on h we can find different f, say f_1 and f_2 positively homogeneous α_1 and α_2 such $\tilde{f} = h_1 \circ f_1$ and $\tilde{f} = h_2 \circ f_2$ (where h_1 and h_2 belong to $\mathcal{M}$). On a convex quadratic function this will particularly imply that linear convergence will hold for a given (γ, q) if there exists $2 \leq \alpha \leq n$ such that $\frac{1}{2}\left(1/\gamma^{\alpha} + \gamma^{\alpha/q}\right) < 1$.

Some further work is needed to prove the integrability of $\mathbf{z} \to \ln \|\mathbf{z}\|$ with respect to the invariant measure of $(\mathbf{Z}_t)_{t \in \mathbb{N}}$ which is established in [18, Lemma 4.2] as a consequence of the geometric drift condition. The almost sure linear convergence can then be derived and the following holds almost surely for all $\mathbf{X}_0$ and for all σ_0

$$\frac{1}{t} \ln \frac{\|\mathbf{X}_t\|}{\|\mathbf{X}_0\|} \xrightarrow[t \to \infty]{} \ln \gamma \left(\frac{q+1}{q} \mathrm{PS} - \frac{1}{q} \right) \tag{7.36}$$

$$\frac{1}{t} \ln \frac{\sigma_t}{\sigma_0} \xrightarrow[t \to \infty]{} \ln \gamma \left(\frac{q+1}{q} \mathrm{PS} - \frac{1}{q} \right) \ , \tag{7.37}$$

where PS is the asymptotic probability of success defined as

$$\mathrm{PS} := \lim_{t \to \infty} P_{\frac{\mathbf{x}}{\sigma}} \left(f(\mathbf{X}_t + \sigma_t \mathbf{U}_{t+1}^1) \leq f(\mathbf{X}_t) \right) = \int 1_{\{f(\mathbf{y}+\mathbf{n}) \leq f(\mathbf{y})\}}(\mathbf{y}, \mathbf{n}) \pi(d\mathbf{y}) p_{\mathcal{N}}(\mathbf{n}) d\mathbf{n} \ , \tag{7.38}$$

where $p_{\mathcal{N}}$ denotes the density of a standard normal distribution (see [18, Theorem 4.5]). Last, the strict negativity of the right-hand side of (7.36) or (7.37) is established [18, Proposition 4.6].

Remark 4. *We are able to establish that both the norm of $\mathbf{X}_t$ and the step-size σ_t converge to zero linearly at the same rate as expressed in* (7.36) *and* (7.37)*. This is compliant with what we observe on simulations (see Figure 7.1, plot on the left).*

7.4 Discussion

The Markov chain methodology presented in this chapter proves its usefulness to establish the linear convergence of comparison-based step-size adaptive randomized search on much wider classes of functions than what was done before. Indeed previous attempts to analyze CB-SARS always focused on much smaller classes of functions. The sphere function was analyzed in [7], [59, 58], and a specific class of convex quadratic functions was also analyzed in [56, 57].

The class of functions where we prove linear convergence for the (1+1)-ES includes all functions deriving from a norm but also non quasi-convex functions. In addition, since if our convergence results hold for f, they also hold for $g \circ f$ for any g in $\mathcal{M}$, we include non-continuous functions in the class of functions where we prove linear convergence.

Interestingly, the condition to obtain a geometric drift reveals that the step-size should increase on a linear function, similar to the condition that was established for the $(1, \lambda)$-ES with self-adaptation. Similarly, when studying the IGO flow trajectories, we also formulated a condition on the step-size increase on the linear function to establish the convergence on C^2 functions [3].

Increasing the step-size on a linear function appears as a natural requirement for step-size adaptive algorithms while some algorithms like the $(1, 2)$-CSA *without cumulation* [35] and the $(1, 2)$-ES with self-adaptation fail to satisfy this condition (see [44] for a thorough analysis of this problem).

Our algorithm framework for the linear convergence study embeds algorithms like CMA-ES without covariance matrix adaptation and without cumulation for the path, i.e. cumulative step-size adaptation (CSA) without cumulation, or the xNES algorithm without covariance matrix adaptation. That is for those algorithms, Proposition 1 holds. For those algorithms however it turns out that the irreducibility, aperiodicity and establishing that compact sets are small sets

is much more intricate to prove than for the $(1+1)$-ES with one-fifth success rule or for comma strategies using self-adaptation. The drift for geometric ergodicity follows however the same lines as for the $(1+1)$-ES with one-fifth success rule. The difficulty comes from the fact that the step-size is a deterministic function of the selected step (this concept was introduced as derandomized self-adaptation), which makes the step-size adaptive algorithm more precise but renders the proof of irreducibility and aperiodicity very difficult.

In an ongoing work with Alexandre Chotard, a full set of general conditions using the underlying control model is established to be able to then easily verify irreducibility, aperiodicity and the fact that compact sets are small sets. This work constitutes a generalization of Chapter 7 of the Meyn and Tweedie seminal book [78]. As a consequence, it should then be easy to finalize the proof of linear convergence of CMA-ES without covariance matrix adaptation and without cumulation for the adaptation of the step-size as well as for xNES without covariance matrix adaptation.

Extension to more complex algorithms, for instance to step-size mechanisms having additional state variables like the step-size mechanism used in CMA-ES or algorithms with an adaptive covariance matrix seems feasible but complex: in this case the drift to prove the geometric ergodicity becomes particularly challenging.

The approach presented here heavily exploits the scaling-invariant property of the class of functions considered. It is not clear how the results can be extended to much more general functions using similar techniques. We believe that then the stochastic approximation approach can be a nice alternative. This is discussed in Chapter 10.

Ideally we would like to be able to obtain theoretically some properties on the convergence rate of the algorithm, for instance its dependency in the dimension or its dependency in the eigenvalues of the Hessian in the case of a convex-quadratic functions. Those types of results seem to be difficult to establish because we do not know much about the invariant measure entering in the definition of the convergence rate.

7.4.1 On the connexion with MCMC

Last we want to stress that the work presented in this chapter underlines a clear connexion between comparison-based stochastic black-box methods and Markov chain Monte Carlo (MCMC) algorithms. MCMC methods are algorithms to sample probability distributions that aim at constructing a stable Markov chain whose invariant distribution is the distribution to be sampled. This latter distribution contains typically some non-singular parts. In contrast, in optimization, we want the algorithm to converge to certain points, i.e. we want the underlying sequence θ_t to converge towards Dirac distributions, such that the Markov chain generated by the optimization algorithm is not stable. However as we have seen, on scaling-invariant functions, a joint potentially stable homogeneous Markov chain associated to the original chain exists (here this chain is $\mathbf{Z}_t = (\mathbf{X}_t - \mathbf{x}^\star)/\sigma_t$. This Markov chain defines an MCMC algorithm associated to the optimization algorithm.

One typical difference however in our context, is that the invariant distribution is unknown. We prove its existence, unicity and deduce some properties like integrability w.r.t. the distribution from (geometric) drift conditions.

Chapter 8

Markov chain analysis for noisy, constrained, linear optimization

Contents

In this chapter, we present three convergence studies of adaptive step-size algorithms that also involve the study of the stability of some Markov chains. The underlying ideas are quite similar to the previous chapter: we are interested in proving the linear convergence or divergence of the algorithm studied. This can be deduced from applying a LLN to a Markov chain that we exhibit. We then carry out the stability study of the Markov chain.

We consider here three different contexts: 1) the optimization of a *linear* function [35, 36], 2) the optimization of a linear function with a *linear constraint* [37] and 3) the optimization of a spherical *noisy* function [60].

While the problems investigated look relatively simple (linear function, spherical function), the study of the underlying Markov chain can turn out to be quite involved. In particular because the study of 1) and 2) are carried out on the step-size mechanism used within CMA-ES, that uses a state variable $\mathbf{p}^\sigma$ to be able to update the step-size.

Moreover, as motivated in the different sections, the problems are per se important to be solved "optimally", so it is important to look carefully at them. More precisely, the belief that adaptive stochastic optimization algorithms should in particular be able to solve simple problems reasonably well has been the driving force behind the development of the CMA-ES algorithm which has turned out to be quite successful.

8.1 Study of the $(1, \lambda)$-ES with cumulative step-size adaptation

The cumulative step-size adaptation (CSA) or path length control is the step-size mechanism used by default in the CMA-ES algorithm. One particularity compared to the step-size adaptive framework of the previous chapter is that an additional state variable, the path, is used to update the step-size, i.e. the state of the algorithm at iteration t is $\theta_t = (\mathbf{X}_t, \sigma_t, \mathbf{p}_t^\sigma)$. This implies that the normalized Markov chain associated on scaling-invariant functions is the couple $(\mathbf{Z}_t, \mathbf{p}_t^\sigma)$ and the analysis of the stability appears to be much more intricate. We however foresee that increasing the step-size on a linear function will be one main condition for the stability. This is one motivation to carefully study CSA on a linear function. Related to that, the linear function models the scenario with (too) small step-size, where hence the algorithm "sees" the objective function very locally and where the step-size should be increased as fast as possible. As we will sketch, this study involves again to investigate the stability of some Markov chains. Those results, summarized in Section 8.1.1, are presented in [35, 36].

8.1.1 Study of the $(1, \lambda)$-ES with CSA on a linear function

The algorithm considered is a $(1, \lambda)$-ES with CSA. It is optimizing a linear function assumed w.l.g. equal to $f(\mathbf{x}) = [\mathbf{x}]_1$ where $[\mathbf{x}]_1$ denotes the first coordinate of the vector $\mathbf{x}$. Given $\mathbf{U}_{t+1} = (\mathbf{Z}_{t+1}^1, \ldots, \mathbf{Z}_{t+1}^\lambda)$ where the sequence $(\mathbf{Z}_{t+1}^i)_i$ follows i.i.d. multivariate normal distributions, the updates of the state variables $\theta_t = (\mathbf{X}_t, \sigma_t, \mathbf{p}_t^\sigma)$ are given by

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \sigma_t \mathbf{Z}_{t+1}^{1:\lambda} \tag{8.1}$$

$$\mathbf{p}_{t+1}^\sigma = (1-c)\mathbf{p}_t^\sigma + \sqrt{c(2-c)}\mathbf{Z}_{t+1}^{1:\lambda} \tag{8.2}$$

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{c}{2d_\sigma}\left(\frac{\|\mathbf{p}_{t+1}^\sigma\|^2}{n} - 1\right)\right) \quad , \tag{8.3}$$

where as before the notation $1{:}\lambda$ denotes the index of the best candidate solution. Note that the last equation is slightly modified compared to the CSA described within the CMA presentation in Section 4.2.1 as here the squared norm (instead of the norm) of the path $\mathbf{p}_{t+1}^\sigma$ is compared to the squared norm (instead of the norm), the path would have under random selection, that is n (see (4.11)). This way the theoretical analysis is simpler. The expected behavior of the algorithm on the linear function is fast linear divergence.

On the linear function $f(\mathbf{x}) = [\mathbf{x}]_1$, the selection changes only the distribution of the first coordinates of the vectors $\mathbf{Z}_{t+1}^i$. More precisely, the first coordinate of $\mathbf{Z}_{t+1}^{1:\lambda}$ is distributed according to the first order statistics of standard normal distributions, i.e. $\mathcal{N}_{1:\lambda}$ and the other coordinates are i.i.d. following standard normal distributions $\mathcal{N}(0, 1)$.

It is then easy to deduce, for the case without cumulation, i.e. $c = 1$, from a simple application of the LLN for independent random variables that the following equation is satisfied for all $\lambda \geq 1$

$$\lim_{t\to\infty} \frac{1}{t} \ln\left(\frac{\sigma_t}{\sigma_0}\right) = \frac{1}{2d_\sigma n}\left(E[\mathcal{N}_{1:\lambda}^2] - 1\right) \text{ a.s.}$$

For $\lambda \geq 3$, the RHS of the previous equation is strictly positive such that *linear divergence* takes place while for $\lambda = 1$ and $\lambda = 2$ an additive unbiased random walk for $\ln(\sigma_t)$ is observed (see [35, 36, Theorem 1]).

With cumulation, i.e. $0 < c < 1$, the path $(\mathbf{p}_t^\sigma)_t$ is a Markov chain. Given that selection is only affecting the first coordinate, we can simply analyze the Markov chain $([\mathbf{p}_t^\sigma]_1)_t$. We can prove without too many difficulties its φ-irreducibility, aperiodicity and the fact that compact sets of $\mathbb{R}$ are small sets for the chain. We then prove that $V(x) = x^2 + 1$ is a drift for geometric ergodicity and that the function $x \mapsto x^2$ is integrable with respect to the stationary distribution of $([\mathbf{p}_t^\sigma]_1)_t$.

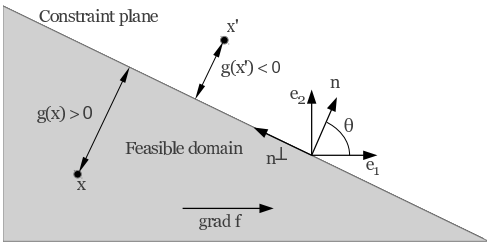

Figure 8.1: Linear function with a linear constraint, in the plane generated by ∇f and $\mathbf{n}$, a normal vector to the constraint hyperplane with angle $\theta \in (0, \pi/2)$ with ∇f. The point $\mathbf{x}$ is at distance $g(\mathbf{x})$ from the constraint.

We finally deduce the *linear divergence* for $c < 1$ with $\lambda \geq 2$ at the rate

$$\frac{1}{t} \ln \left(\frac{\sigma_t}{\sigma_0} \right) \xrightarrow[t \to \infty]{a.s.} \frac{1}{2 d_\sigma n} \left(2(1-c) E[\mathcal{N}_{1:\lambda}]^2 + c(E[\mathcal{N}_{1:\lambda}^2] - 1) \right) \quad . \tag{8.4}$$

Divergence also holds when considering the expectation in the LHS of the previous equation [35, 36, Theorem 3]. Equation 8.4 quantifies the divergence rate of the CSA on a linear function as a function of c, λ and n. We have also studied the variance of $\ln(\sigma_{t+1}/\sigma_t)$ and found out that keeping $c < 1/n^{1/3}$ ensures that the standard deviation of $\ln(\sigma_{t+1}/\sigma_t)$ becomes small enough in front of $\ln(\sigma_{t+1}/\sigma_t)$ when dimension goes to infinity confirming that the default cumulation parameter which is smaller than $1/\sqrt{n}$ is a reasonable choice.

8.1.2 Study of the $(1, \lambda)$-ES with CSA using resampling on a constraint problem

We consider here the problem of optimizing a linear function with a linear constraint with a $(1, \lambda)$-CSA using in addition a resampling mechanism to ensure that the candidate solutions are within the bounds. More precisely we want to

$$\begin{aligned} \text{maximize} \quad & f(\mathbf{x}) = -\mathbf{x}.\mathbf{n} = [\mathbf{x}]_1 \quad \text{subject to} \\ & g(\mathbf{x}) = -[\mathbf{x}]_1 \cos\theta - [\mathbf{x}]_2 \sin\theta \geq 0 \quad . \end{aligned} \tag{8.5}$$

where $\mathbf{n}$ is a vector normal to the constraint and where $\theta \in (0, \pi/2)$ (see Fig 8.1). Note that the point $\mathbf{x}$ is at distance $g(\mathbf{x})$ from the constraint. The update equations for the CSA with resampling boil down to (8.1), (8.2), (8.3) where the distribution of the vector $\mathbf{U}_{t+1} = (\mathbf{Z}_{t+1}^1, \ldots, \mathbf{Z}_{t+1}^\lambda)$ is such that each $\mathbf{Z}_{t+1}^i$ results from resampling multivariate normal distributions till they lie within the feasible domain. In other words in the direction $\mathbf{n}$, the distribution of $\mathbf{Z}_{t+1}^i$ is a truncated Gaussian and it follows standard normal distributions in directions orthogonal to $\mathbf{n}$.

We define δ_t as the normalized distance to the constraint, i.e.

$$\delta_t = g(\mathbf{X}_t)/\sigma_t \quad .$$

and we show that $(\delta_t, \mathbf{p}_t^\sigma)$ is a Markov chain ([37, Proposition 5]). Given that

$$\frac{1}{t} \ln \frac{\sigma_t}{\sigma_t} = \frac{c}{2 d_\sigma} \left(\frac{1}{nt} \sum_{i=0}^{t-1} \|\mathbf{p}_i^\sigma\|^2 - 1 \right) \quad , \tag{8.6}$$

the study of the stability of the Markov chain $(\delta_t, \mathbf{p}_t^\sigma)$ can give us the linear divergence or convergence of the algorithm. We have carried out this stability study for the case where $c = 1$ and proven the φ-irreducibility, aperiodicity, positivity and geometric ergodicity of the chain that allow us to conclude to the linear *divergence or convergence* of the algorithm. We refer to [37,

Theorem 2] for the explicit expression of the divergence (or convergence) rate that does not have a very comprehensive expression and is thus not reproduced here.

To find out whether divergence or convergence is happening, we have carried out numerical simulations of the convergence rate and found out that CSA diverges (i.e. the algorithm works as we want) for small enough cumulation parameter or large enough population size. However, small values of the constraint angle increase the difficulty of the problem arbitrarily such that we cannot find values of c and λ where the algorithm diverges.

8.2 Linear convergence or divergence of a $(1+1)$-ES in noisy environment

We consider a noisy optimization problem, that is for each point of the search space $\mathbf{x} \in \mathbb{R}^n$ the objective function $f(\mathbf{x})$ is perturbed by a random variable, i.e for a given $\mathbf{x}$, we can observe a distribution of possible objective function values. We investigate a certain class of noisy problems, which use the so-called *multiplicative noise*, where the noiseless objective function $f(\mathbf{x})$ is perturbed by the multiplication with a noise term independent of $\mathbf{x}$ and thus of $f(\mathbf{x})$. The plain multiplicative-noisy objective function $\hat{f}$ reads

$$\hat{f}(\mathbf{x}) = f(\mathbf{x})\xi \tag{8.7}$$

The noise random variable, ξ, is sampled independently at each new evaluation of a solution point.

A typical goal in noisy optimization is to converge to the minimum of the averaged value of the observed random variable $\hat{f}$. If the expected value of the noise in (8.7) is positive, this means minimizing f. We more precisely consider the following noisy objective function

$$\hat{f}(\mathbf{x}) = g(\|\mathbf{x}\|^\alpha \xi) \tag{8.8}$$

where $g \in \mathcal{M}$. We assume that the law of ξ has a probability density function denoted p_ξ. We also assume that the support of p_ξ is the range $]m_\xi, M_\xi[$ where $-\infty \leq m_\xi < M_\xi \leq +\infty$ and $m_\xi \neq 0$.

We investigate the $(1+1)$-ES with step-size proportional to the optimum and show essentially two results:

- First we prove that divergence or convergence of the algorithm can happen even when the expected objective function value is a positive function with a unique minimum. We prove that the divergence versus convergence is solely determined by the sign of m_ξ the left bound of the interval supporting the noise distribution.
- Second we prove that a linear behavior takes place, i.e. linear convergence takes place if $m_\xi > 0$ and linear divergence takes place if $m_\xi < 0$. This result is a theoretical statement of the robustness of ESs in noisy environments. The proof of this linear behavior involves the stability study of a Markov chain.

8.2.1 The algorithm considered

We consider $(\mathbf{U}_t)_{t\in\mathbb{N}}$ a sequence of i.i.d. random vectors distributed according to $\mathcal{N}(0, I_d)$ and $(\xi_t)_{t\in\mathbb{N}}$ a sequence of random variables distributed according to p_ξ and independent from the sequence $(\mathbf{U}_t)_{t\in\mathbb{N}}$. We consider $\mathbf{X}_0 \in \mathbb{R}^n$ also independent from $(\mathbf{U}_t)_{t\in\mathbb{N}}$ and $(\xi_t)_{t\in\mathbb{N}}$ such that $\|\mathbf{X}_0\| > 0$ almost surely. The objective function value associated to $\mathbf{X}_0$ equals $g(\|\mathbf{X}_0\|^\alpha \xi_0)$[1].

We introduce the sequence O_t as the sequence of selected noise that is defined iteratively starting from $O_0 = \xi_0$.

We assume a step-size proportional to the distance to the optimum, that is $\sigma_t = \sigma\|\mathbf{X}_t\|$ such that

$$\mathbf{X}_{t+1} = \begin{cases} \mathbf{X}_t + \sigma\|\mathbf{X}_t\|\,\mathbf{U}_{t+1} & \text{if } g\left(\left\|\mathbf{X}_t + \sigma\|\mathbf{X}_t\|\mathbf{U}_{t+1}\right\|^\alpha \xi_{t+1}\right) < g\left(\|\mathbf{X}_t\|^\alpha O_t\right) \\ \mathbf{X}_t & \text{otherwise}\,, \end{cases} \tag{8.9}$$

[1] All the random vectors are assumed to be defined on a same probability space.

and the accepted noise O_{t+1} of the new parent $\mathbf{X}_{t+1}$ obeys:

$$O_{t+1} = \begin{cases} \xi_{t+1} & \text{if } g\left(\left\|\mathbf{X}_t + \sigma\|\mathbf{X}_t\|\mathbf{U}_{t+1}\right\|^\alpha \xi_{t+1}\right) < g\left(\|\mathbf{X}_t\|^\alpha O_t\right) \\ O_t & \text{otherwise .} \end{cases} \tag{8.10}$$

Note that since g preserves the ordering it can be dropped in the acceptance criteria. The sequence O_t can be written in a more compact manner as

$$O_{t+1} = O_t + (\xi_{t+1} - O_t)1_{\{\|\mathbf{X}_t/\|\mathbf{X}_t\| + \sigma\mathbf{U}_{t+1}\|^\alpha \xi_{t+1} < O_t\}} \ .$$

Then $(O_t)_{t\in\mathbb{N}}$ is an homogeneous Markov chain with same initial law and transition kernel as

$$Z_{t+1} = Z_t + (\xi_{t+1} - Z_t)1_{\{\|e_1 + \sigma\mathbf{U}_{t+1}\|^\alpha \xi_{t+1} < Z_t\}} \tag{8.11}$$

with Z_0 distributed according to p_ξ [60, Proposition 1].

8.2.2 Linear convergence or divergence

To investigate the linear convergence or divergence, we study the same quantity as previously, namely

$$\frac{1}{t}\ln\frac{\|\mathbf{X}_t\|}{\|\mathbf{X}_0\|} = \frac{1}{t}\sum_{k=0}^{t-1}\ln\left(\left\|\frac{\mathbf{X}_k}{\|\mathbf{X}_k\|} + \sigma\mathbf{U}_{k+1}1_{\left\{\left\|\frac{\mathbf{X}_k}{\|\mathbf{X}_k\|}+\sigma\mathbf{U}_{k+1}\right\|^\alpha \xi_{k+1} < O_k\right\}}\right\|\right) \ . \tag{8.12}$$

The RHS of the previous equality can be expressed using rotational invariance of the multivariate normal distribution with the Markov chain (Z_t) introduced above, namely

$$\frac{1}{t}\ln\frac{\|\mathbf{X}_t\|}{\|\mathbf{X}_0\|} = \frac{1}{t}\sum_{k=0}^{t-1}\ln\|e_1 + \sigma\mathbf{U}_{k+1}1_{\{\|e_1+\sigma\mathbf{U}_{k+1}\|^\alpha \xi_{k+1} < Z_k\}}\|$$

where the previous equality holds in distribution. To take the limit when t goes to infinity in the previous equation, we investigate the Markov chain (Z_t) and prove that it is positive and Harris recurrent by proving that it is uniform ergodic [60, Proposition 2]. We then deduce the convergence of the previous equation for t to infinity and obtain the following theorem.

Theorem 9 ([60, Theorem 6])**.** *The (1+1)-ES defined in* (8.9) *(and* (8.10)*) minimizing the noisy sphere (*(8.8)*) converges almost surely to zero if $m_\xi > 0$ and diverges almost surely to infinity when $-\infty < m_\xi < 0$. For $m_\xi \neq 0$, let γ be defined as*

$$\gamma := \int E\left(\ln\|e_1 + \sigma\mathcal{N}_0 1_{\{\|e_1+\sigma\mathbf{U}_1\|^\alpha \xi_1 \leq z\}}\|\right) d\mu(z) \tag{8.13}$$

where μ is the invariant probability measure of the Markov chain $(Z_t)_t$ ((8.11)*). Then γ is well defined, finite and the algorithm converges (or diverges) log-linearly in the sense that:*

$$\frac{1}{t}\ln\|\mathbf{X}_t\| \to \gamma \tag{8.14}$$

holds in probability. Moreover, the convergence (or divergence) rate γ is strictly negative if $m_\xi > 0$ and strictly positive if $m_\xi < 0$.

Chapter 9

A glimpse on other topics

Contents

This chapter gathers some of my contributions that are either not directly related to single-objective optimization using adaptive comparison-based algorithms or not theoretical.

In a first part, I give an overview of my work in the context of multi-objective optimization where the goal is to optimize simultaneously several conflicting objectives. Those contributions are related to the study of the so-called *hypervolume*, an indicator to measure the quality of sets of solutions widely used in the domain of Evolutionary Multi-objective Optimization (EMO). The overview given is related to the publications [10], [22], [21], [8], [9].

In a second part, I present some of my contributions in the domain of benchmarking of continuous black-box algorithms. One main motivation has been to develop better benchmarking methodologies and thus improve the standards in how benchmarking is done in the domain of continuous black-box optimization. This work is a long term project that was somehow started in 2005 with the participation to a benchmarking special session [16, 15] and that we have intensified with the creation of a benchmarking platform, COCO—developped since 2008—and of a benchmarking workshop, the Black-Box Optimization Benchmarking (BBOB) workshop, that we have been organizing recurrently since 2009. This part is related to [47, 48, 46] but also to (so far unpublished) work in preparation.

Last I sketch some work related to the optimal placement of oil wells. Those contributions are linked to the PhD thesis of Zyed Bouzarkouna in the context of a collaboration with the French Institute for Petrol (IFP). The related publications are [32, 29, 30, 31].

Note that while those contributions are presented in the end of the manuscript, I do not consider those contributions as minor or less interesting. Actually the work on multi-objective optimization has been quite influential as witnessed by the fact that our two main publications [10] and [22] have overall 148 citations according to Google Scholar[1]. Also our work on benchmarking has already some impact: 174 papers on Google Scholar are referring to "Black-Box optimization benchmarking (BBOB)" and the Google Scholar citations of the main papers describing our benchmarking platform [47] [48] [46] have 393, 56, and 230 citations respectively.

[1] As of March 2015.

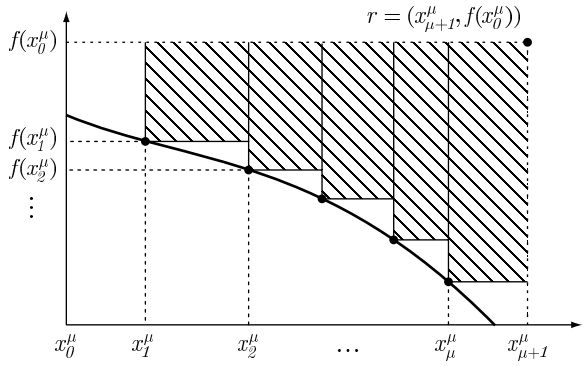

Figure 9.1: Hypervolume indicator computation for μ solutions $(x^\mu_1, f(x^\mu_1)), \ldots, (x^\mu_\mu, f(x^\mu_\mu))$ and the reference point $r = (r_1, r_2)$ in the biobjective case.

9.1 Multi-objective optimization

In multi-objective optimization, the goal is to optimize simultaneously several (conflicting) objectives. For instance one can be interested to reduce the cost of the design of a product while maximizing its robustness. The "optimal solutions" that are then thought are the set of best compromises, that is, informally speaking, the set of solutions that cannot be improved along one objective without degrading in at least another one. Formally, let us consider the following (vector-valued) function $\mathcal{F} : \mathbf{x} \in \mathbb{R}^n \to \mathcal{F}(\mathbf{x}) = (\mathcal{F}_1(\mathbf{x}), \ldots, \mathcal{F}_m(\mathbf{x})) \in \mathbb{R}^m$. The space $\mathbb{R}^m$ being then called the *objective space* while like in the single-objective space, $\mathbb{R}^n$ is the search space. We assume without loss of generality that we want to minimize each $\mathcal{F}_i$ simultaneously.

The weak Pareto-dominance relation is given by $\preceq$ defined for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ as

$$\mathbf{x} \preceq \mathbf{y} \Leftrightarrow \mathcal{F}_i(\mathbf{x}) \leq \mathcal{F}_i(\mathbf{y}) \text{ for all } i \ , \tag{9.1}$$

i.e. $\mathbf{x}$ is not worse than $\mathbf{y}$ on all objectives. The optimal solutions (Pareto optima) for $\mathcal{F}$ are given by the minimal elements of the ordered set $(\mathbb{R}^n, \preceq)$. The image of the *Pareto-set* in the objective space is the so-called *Pareto-front*.

Evolutionary Multi-Objective algorithms—stochastic search algorithms to approach multi-objective problems in the EC context— mainly focus on approximating the Pareto-optimal set. Typically they exploit the population-based framework of the algorithms and attempt to make the population converge towards the Pareto-optimal set. One approach to guide this convergence is to use the so-called *hypervolume* as a quantitative measure of the quality of a set of solutions [94]. This hypervolume is simply the area comprised between the set of solutions and a *reference point* (or reference set), more precisely see Figure 9.1 for the visualisation of the volume in the case of a bi-objective problem. Hypervolume-based EMO algorithms look for a set of solutions maximizing the hypervolume.

One advantage of the hypervolume indicator, as opposed to other indicators also used to measure quality of sets, is that it is compliant with the Pareto-dominance relation, i.e. if a set dominates another one[2], its hypervolume will be larger. Hence it became one of the most used indicators in indicator-based EMO algorithms.

Many conjectures on how the hypervolume indicator is influencing the final repartition of the set of solutions found were formulated before. For instance, Zitzler and Thiele [94] indicated that, when optimizing the hypervolume in maximization problems, "convex regions may be preferred to concave regions", which is also stated in [73], whereas Deb et al. [39] argued that "[...] the hypervolume measure is biased towards the boundary solutions". Knowles and Corne observed sets of

[2]A set A dominates a set B if for all points in B, there exists a point in A that dominates it in the sense of (9.1) where in addition at least one inequality for the dominance is strict.

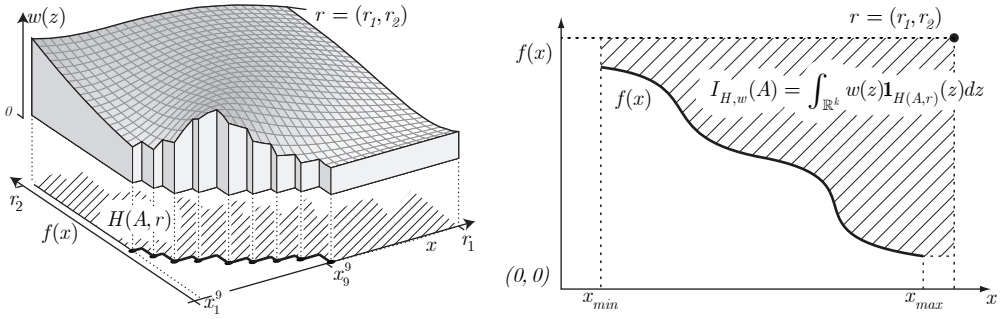

Figure 9.2: The hypervolume indicator $I_{H,w}(A)$ corresponds to the integral of a weight function $w(z)$ over the set of objective vectors that are weakly dominated by a solution set A and in addition weakly dominate the reference point r (hatched areas). On the left, the set A consists of nine objective vectors whereas on the right, the infinite set A can be described by a function $f : [x_{\min}, x_{\max}] \to \mathbb{R}$. The left-hand plot shows an example of a weight function $w(z)$, where for all objective vectors z that are not dominated by A or not enclosed by r the function w is not plotted, such that the weighted hypervolume indicator corresponds to the volume of the gray shape (Figures and caption extracted from [22]).

solutions resulting from maximizing the hypervolume indicator "seems to be 'well-distributed' " [67, 68, 42].

Additionally, an important question arising in practice is how to choose the reference point. In particular, extremes of the Pareto-front should preferably be included in the set.

In this context, we have analyzed for bi-objective problems, that is $m = 2$, the properties of a set of μ points maximizing the hypervolume, referred to as *optimal μ-distributions*, that is the set that hypervolumed-based EMO attempt to approach. We have assumed that the Pareto-front (located in the objective space) is described by a non-increasing continuous function $x \in [x_{\min}, x_{\max}] \mapsto f(x)$.

We have established the exact location of optimal μ-distributions in the case of a linear front. For f derivable on $]x_{\min}, x_{\max}[$ we have derived the expression of the density of points for μ to infinity. More precisely we have shown that this density is proportional to $\sqrt{f'(x)}$ contradicting hence the previous beliefs that convex regions of the Pareto-front are preferred over concave or that there exists a bias towards boundary solutions. We have then addressed the practical question of the choice of the reference point. For fronts characterized via f derivable on $]x_{\min}, x_{\max}[$, we provide a bound on the location of the reference point to be able to enclose the extreme of the fronts. We also show that for some fronts, the extremes cannot be included (when ($f'(x_{\max}) = 0$ the right extreme cannot be included and when $f'(x_{\min}) = -\infty$, the left extreme cannot be included). Those results are presented in [10] and were generalized to the case of the weighted hypervolume where the volume is computed with respect to a pre-defined density on the objective space (see Figure 9.2), the motivation being to allow users to set preferences beforehand on the regions they are interested to see solutions [22].

We have also extended part of the results to the case with three objectives [21]. Last we have exploited the weighted hypervolume idea to show how to practically articulate user preferences within an indicator-based algorithm [8, 9].

9.2 Benchmarking

While theory is helping to understand better how algorithms work or to design new algorithms (as we have illustrated in Chapter 6), there are some clear limitations in terms of how much theory can tell as far as performance of algorithms is concerned. On the one hand, the class of functions where one can prove convergence is limited, on the other hand the convergence proofs are rarely

accompanied with exact convergence rates (rather bounds on the convergence rates are proven).

However, it is central to assess performance of algorithms *quantitatively* on functions that represent the typical difficulties related to real-world problems the algorithms are supposed to solve. In a second stage, performance of algorithms can be compared to understand globally the strengths and weaknesses of different approaches.

For this, it is necessary to ressort to *benchmarking* of algorithms that consist in running the algorithms on *well-chosen* test functions and extracting data to assess and compare performances.

It turns out however that it is not trivial to do proper benchmarking of algorithms. Many possible biases can be introduced when benchmarking. It was already underlined by J. N. Hooker in 1995 in the paper "Testing Heuristics: We Have It All Wrong" [55].

In this context, some of my research since my PhD has been focusing on trying to improve how benchmarking in the black-box optimization context is done. I will sketch below the main scientific aspects we have been concerned with. The diffusion of the practice we wanted to establish in terms of benchmarking has been done through the development of the benchmarking platform COmparing Continuous Optimizers (COCO)[3]—developed since 2008—that automatizes the benchmarking of optimization algorithms from running experiments to post-processing them in order to display graphs and tables presenting the benchmarking results. We have then been organizing with this platform the Black-Box Optimization Benchmarking (BBOB) workshops[4] (this year the 5^{th} and 6^{th} editions of the workshops will take place) where participants are encouraged to benchmark their favorite algorithm with the COCO platform, submit papers with the post-processed data (also produced with the platform) and submit their benchmarking data-set such that it can be made publicly accessible. We have been able to collect so far 120+ algorithm data-sets.

The important scientific aspects, we have been concerned with when developing the COCO / BBOB framework for benchmarking, are summarized below.

On the choice of the test functions. The choice of the test functions is crucial. However, too often test functions are chosen because they are easy to construct. As a result, we often see bias in the test-suite used to benchmark algorithms. We have already mentioned the bias towards separable functions in Chapter 5, but we have observed also bias towards small dimensional test functions or towards convex functions in the CUTEr test-suit for instance. One aspect to realize is that if performance is aggregated over functions, a bias towards a certain type of functions (small dimensional for instance) put forwards algorithms that are especially good for solving this category of problems. While this remark is simple, it seems to be generally overlooked when analyzing benchmarking results. Our approach to design the BBOB test functions has been that the test-suite should represents the typical difficulties encountered in real world applications (ill-conditioning, non-separability, multi-modality with weak or strong global structure, noise). The functions are *simple enough* such that the difficulties behind each function can be understood but at the same time *challenging* for algorithms. Scientific questions can be answered from each test function. Overall, we have designed a first test-suit presenting 24 noiseless functions [47] and a second one presenting 30 noisy functions [48]. In addition the test-suite proposed is scalable with respect to the dimension, i.e. each test function is defined for any possible dimension parameter [47, 48].

On measuring performance. In the black-box setting, the running time of an algorithm is measured in terms of *number of function evaluations* (or calls to the black-box) as opposed to real CPU time. Indeed, in the context of black-box optimization (where the objective function can be the outcome of large numerical simulations), it is reasonable to assume that the prominent cost is related to the function evaluations and not to internal CPU time of the algorithm. Using the number of function evaluations instead of the CPU time presents the main advantage that the number of function evaluations is a measure independent of the programming language or on the

[3] `http://coco.gforge.inria.fr/doku.php`
[4] `http://coco.gforge.inria.fr/doku.php`

skills of the programmer. As denounced in [55], using CPU time as measurement can lead to focus too much attention and time in implementation details or in implementing in "fast" programming languages and distract researchers from doing research on algorithms.

Given that the running-time is measured in function evaluations, we advocate that performance measures should be quantitative, with a well-interpretable meaning. This has an impact on how data of experiments are collected. Given a run carried out on a test function, we can decide to

vertical view: either collect at different fixed budgets the function value proposed by the search algorithm or

horizontal view: fix a set of function-value targets and collect the number of function evaluations needed to reach those targets.

The first approach ("vertical view") is easier to implement and does not require to handle the fact that some target values are not reached. It is hence often the chosen approach. It means that a possible performance measure is based on f-values at different budgets. However the meaning of an f-value is not quantitative, cannot be easily interpreted and varies from one function to the next one. We therefore collect data using the "horizontal view", i.e. collect running times to reach a certain f-target.

Our main measure of performance based on the collected data is the expected running time ERT associated to a (function, target) couple. Formally, assume we have an algorithm with a strictly positive probability p_s to reach a given target, we consider the expected running time of the algorithm which is restarted until success (hence the restart algorithm has a probability one to converge). The expected running time of the restart algorithm equals

$$E[\mathrm{RT}] = \left(\frac{1}{p_s} - 1\right) E[\mathrm{RT}_{\mathrm{unsuccessful}}] + E[\mathrm{RT}_{\mathrm{successful}}] \ , \tag{9.2}$$

where $\mathrm{RT}_{\mathrm{unsuccessful}}$ is a random variable that models the running time for unsuccessful runs and $\mathrm{RT}_{\mathrm{successful}}$ the running time for successful runs.

Given a finite number of independent runs, N_{runs}, of an algorithm on a given test function, and a collection of $N_{\mathrm{unsuccessful}}$ realizations of the random variables $\mathrm{RT}_{\mathrm{unsuccessful}}$ and $N_{\mathrm{successful}}$ realizations of $\mathrm{RT}_{\mathrm{successful}}$ such that $N_{\mathrm{runs}} = N_{\mathrm{unsuccessful}} + N_{\mathrm{successful}}$, a natural estimator for $E[\mathrm{RT}]$ is

$$\mathrm{ERT} = \frac{\sum RT^i_{\mathrm{unsuccessful}} + \sum RT^i_{\mathrm{successful}}}{N_{\mathrm{successful}}} = \frac{\#\text{Total number of evaluations}}{N_{\mathrm{successful}}} \ .^{5} \tag{9.6}$$

On displaying performance. In the COCO / BBBO setting, we have a large amount of collected data. For instance in the noiseless case, for each of the 24 functions, we collect for a large amount of targets (around 50), the running times of 15 different runs for 6 different dimensions ($n = 2, 3, 5, 10, 20, 40$). It is hence crucial to display the results in a meaningful way and have other alternatives than providing tables of numbers. One series or useful graphs are scaling graphs showing for each function, ERT as a function of the dimension for 7 different targets (see Figure 9.3 and also in Figure 1 of [11] available at `http://researchers.lille.inria.fr/~brockhof/publicationListFiles/abh2010k.pdf`). For comparing two algorithms scatter plots

[5] The estimate (9.4) comes naturally from

$$E[\mathrm{RT}] \approx \left(\frac{N_{\mathrm{runs}}}{N_{\mathrm{successful}}} - 1\right) \frac{\sum RT^i_{\mathrm{unsuccessful}}}{N_{\mathrm{runs}} - N_{\mathrm{successful}}} + \frac{\sum RT^i_{\mathrm{successful}}}{N_{\mathrm{successful}}} = \tag{9.4}$$

$$\left(\frac{N_{\mathrm{runs}} - N_{\mathrm{successful}}}{N_{\mathrm{successful}}}\right) \frac{\sum RT^i_{\mathrm{unsuccessful}}}{N_{\mathrm{runs}} - N_{\mathrm{successful}}} + \frac{\sum RT^i_{\mathrm{successful}}}{N_{\mathrm{successful}}} \tag{9.5}$$

$$= \frac{\#\text{Total number of evaluations}}{N_{\mathrm{successful}}} \tag{9.6}$$

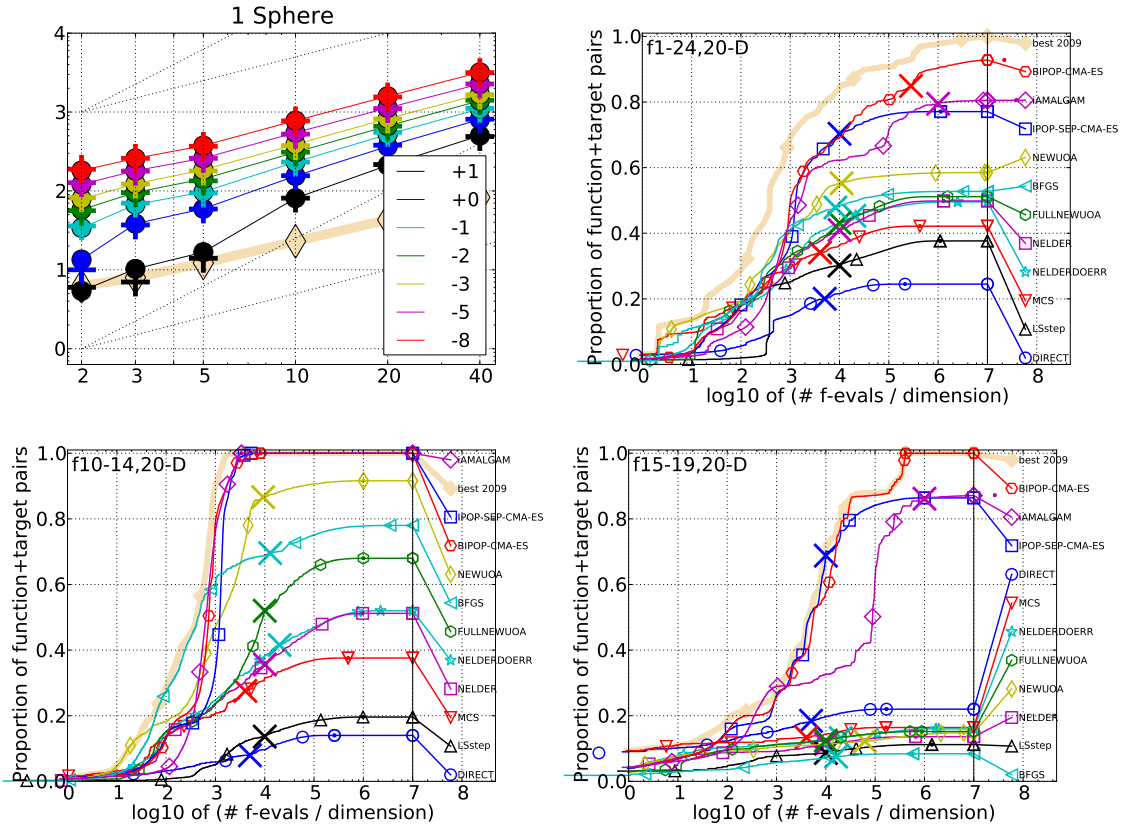

Figure 9.3: **Top Left:** Example of display of performance on single functions using ERT. $\log_{10}$ of ERT divided by dimension versus dimension on the sphere function for different targets ranging from 10^{1} to 10^{-8}. **Top Right:** Empirical Cumulative Distribution Function (ECDF) aggregated overall all functions and all targets of the noiseless BBOB test-suite in 20-D for 11 different algorithms (name is displayed on the right). **Low Left:** ECDF of the same algorithms in 20-D aggregated over functions with a high condition number only. **Low Right:** ECDF of the same algorithms in 20-D aggregated over multi-modal functions only.

of ERT are also useful (see for instance Figure 2 in [33] available at `http://researchers.lille.inria.fr/~brockhof/publicationListFiles/bah2012c.pdf`).

An other type of useful display to aggregate information is to plot the empirical cumulative distribution (ECDF) of the run-length. One peculiarity of the COCO framework is that we aggregate results over several targets but not over dimension. ECDF plots allow to compare in a same graphs various algorithms. We plot ECDF graphs over all functions and over subset of functions having specific properties (see Figure 9.3).

9.3 Application to optimal placement of oil wells

We describe in this last part an industrial black-box application and sketch the different contributions that were done to address it. The problem consists in finding optimal (petroleum) well placements in a given reservoir and was investigated by the French Institute for Petrol through the thesis of Zyed Bouzarkouna. Different reservoir simulators able to predict for a certain set of well configurations the amount of oil, gaz and water that could be extracted are available. The simulators are typically some real black-box because they are either commercial solvers for which only an executable of the code is available or much too complex such that useful information

can be extracted. Each evaluation of a configuration of well is taking between 20 minutes to several hours such that the formulated black-box optimization problem is *expensive*. The working hypothesis is to use *non-conventional* wells, that typically have several connected non-vertical branches, in order to increase the quantity of hydrocarbon that can be recovered (while typical conventional wells are composed of a single vertical branch). Those non-conventional wells have a higher drilling cost that is taken into account in the formulation of the optimization problem. The optimization problem is to place several wells in a reservoir simulator and maximize the Net Present Value associated. The optimization landscape is typically rugged due to heterogeneities in the geology of the reservoir. The CMA-ES algorithm is thus a good candidate algorithm to tackle the problem that was previously addressed with Genetic Algorithms (GAs) [32].

However, because of the specifically expensive setting, it is natural to address the problem using surrogate or hybrid approaches where a meta-model of the objective function is learned, this model is used to save some expensive function evaluations [32]. The algorithm that was used is the local-meta-model CMA-ES (lmm-CMA). The core of the lmm-CMA algorithm is CMA-ES, however some evaluations on the expensive function are saved by doing the following procedure. Given a large enough archive of (points, function value of the points), for a new query-point sampled within CMA that needs to be evaluated, a local quadratic models of the objective function is learned by using a weighted regression using the points from the archive. The metric used to choose the points closed to query-point and the weights is the Mahalanobis distance associated to the current covariance matrix of CMA-ES. This meta-model building is repeated for each of the λ sampled points of CMA-ES. To know whether the function-value prediction of the meta-model is good, a fraction of the λ points are evaluated on the expensive objective function and the meta-model building procedure is restarted using the freshly enriched archived. If the predicted ranking of the points with the new meta-models and points evaluated on the expensive function did not change, the ranking predicted is considered as good enough and a next iteration of CMA-ES is started [66] [29, 14].

The particularity of the optimization problem was taken into account in a variant of lmm-CMA. More precisely, given that the wells are placed in the same reservoir it is reasonable to assume that the objective function to optimize is partially separable, that is, the objective function can be written as the sum of sub-functions where each sub-function depends on the local parameters of a single well plus an additional global parameter (typically encoding the distance between two wells). It has been possible to design a variant of lmm-CMA using meta-models having this specific objective function structure [30] and applied it successfully to the well placement optimization problem [31].

Chapter 10

Discussion and perspectives

Black-box continuous optimization methods are needed in various domains in academy or industry. Those past years have known a resurgence of interest for such methods in the mathematical programming community. The methods in this field, presented under the name derivative-free optimization methods, are mostly deterministic [38]. In contrast, this manuscript has presented *stochastic* continuous black-box methods, tailored to tackle challenging numerical optimization problems with an emphasis on *comparison-based* adaptive methods whose most notable example is the covariance matrix adaptation evolution strategy (CMA-ES). Both the stochastic component and the comparison-based property confer some robustness to the methods that is particularly useful when rugged or non-convex problems needs to be solved.

While introduced in an engineering and computer-science context, the methods of interest in the manuscript and particularly CMA-ES have strong mathematical foundations. Those foundations were often discovered after the introduction of the methods. We have in particular sketched the connexion with information geometry and detailed the connexion with Markov chain Monte Carlo (MCMC) methods. We have also emphasized the importance of invariance.

One central theoretical question when studying optimization algorithms is whether they converge and at which rate. Linear convergence observed on wide classes of functions for comparison-based adaptive algorithms like CMA-ES is an important property of the methods.

We have presented a general methodology in the context of step-size adaptive algorithms to prove the linear convergence on so-called scaling-invariant functions. This methodology exploits invariance properties (scale and translation invariance) of the algorithm to exhibit a Markov chain candidate to be "stable". Using tools particularly developed and used in the context of MCMC, we have presented a proof of the *linear convergence* for the $(1+1)$-ES with one-fifth success rule—a natural method introduced by several authors independently already in the 70's. The class of functions where the linear convergence holds includes non-quasi-convex and non-continuous functions.

We want to argue that those theoretical results have a strong practical meaning: they allow to handle real algorithms without any simplification assumption as opposed to some results obtained with stochastic approximation methods that rely on predefined gain sequences, which seem unrealistic in practice [88].

We have also illustrated that the theory of Markov chains on a continuous state space is useful to study comparison-based algorithms in various contexts: optimization without the presence of noise, constrained optimization and noisy optimization. While the algorithm analyzed in the context of noise is assuming an optimal step-size, we still argue that the preservation of the linear convergence property in the case of multiplicative noise is a strong theoretical hint of the robustness of comparison-based stochastic adaptive algorithms.

We have also presented convergence bounds for specific algorithm frameworks. Those bounds are linked to the (approximative) theory that accompanied the development of Evolution Strategies since its introduction, namely the *progress rate* theory. They are quantitative and we have

explained the relative tightness of the bounds. In addition, asymptotic explicit estimates of the bounds w.r.t. dimension can be rigorously obtained. We have then described how such bounds are useful for algorithm design.

So what is next? We strongly believe that the Markov chain methodology and the connexion with MCMC methods to prove linear convergence presented in Chapter 7 can be exploited *much further* to give convergence proofs of more algorithms:

- Under the same assumption that the state of the algorithm is reduced to $\theta_t = (\mathbf{X}_t, \sigma_t)$, we would like to cover algorithms like CSA (without cumulation) or xNES (where the covariance matrix is σ_t^2 times identity) where the step-size change is tightly connected to the steps used to update the mean vector $\mathbf{X}_t$ (as opposed to self-adaptive algorithms [7]). It turns out that proving the irreducibility and aperiodicity of the normalized chain is then becoming very difficult while finding and proving a drift for geometric ergodicity does not appear to pose major problems. It therefore seems that we need to develop specific tools for Markov chain models that embed the chains we are investigating. One approach to do so, that constitutes an on-going work with Alexandre Chotard, consists in extending the connection of irreducibility and aperiodicity with the underlying control model as done in Chapter 7 of [78] for chains of the form
$$\theta_{t+1} = F(\theta_t, \alpha(\theta_t, \mathbf{U}_{t+1}))$$
where F is C^1, $(\mathbf{U}_t)_t$ is i.i.d. and α is a deterministic function of the state θ_t and the independent random vector $\mathbf{U}_{t+1}$ that models in our case the selection process.

- It is also natural to extend the approach to algorithms with more state variables for instance for CSA with cumulation where the path needs to be taken into account or for CMA-ES where the state includes a covariance matrix. The construction of the normalized Markov chain to be studied seems relatively straightforward while establishing drift conditions becomes very challenging. Indeed, establishing a drift condition requires to prove the negativity of the drift function "outside a small set": while the study of what is going on outside a small set in the case of $\theta_t = (\mathbf{X}_t, \sigma_t)$ boils down to the study of a simple one-dimensional function, controlling the drift for CMA or CSA outside a small set requires to handle simultaneously different scenarios, each of them being less straightforward.

  Still, we strongly believe that the functioning of CMA-ES is tightly connected to the Markov chain methodology that relies on invariance and stability of an underlying Markov chain constructed by exploiting invariance properties of the algorithm. We foresee in particular that affine invariance will be central for the CMA convergence proof. We also understand that the stability is related to proper learning rates (i.e. the algorithm is not stable for too large learning rates).

Finally, the ODE method or stochastic approximation framework [74, 28, 69] has not been really explored so far to analyze the convergence of comparison-based adaptive stochastic methods. Some first attempts in that direction were however presented by Yin et al. [92, 93]. They have analyzed a step-size adaptive evolution strategy. Nevertheless, they have only considered the mean vector as state variable of the algorithm and imposed the variance to be equal to the gradient of the objective function. Hence their analyzed algorithm departs significantly from comparison-based step-size adaptive algorithms. In addition, they have assumed that the learning rate for the mean update decreases to zero and can therefore not obtain linear convergence (Theorem 5.2 of [93]).

We believe that it is however possible to use the ODE method for proving the linear convergence of step-size adaptive ESs by encoding the "real" state of the algorithm, namely in a first time $\theta_t = (\mathbf{X}_t, \sigma_t)$. We think it is possible to obtain results with a fixed learning rate, that will however need to be chosen small enough. The approach will rely on the one hand on studying the solutions of the underlying mean field ODE similarly to what was done in [3] and on the other hand on adapting current tools to control the error between the solution of the ODE and the stochastic trajectory. This is an ongoing work together with Youhei Akimoto.

Chapter 11

Appendix

11.1 Proof from invariance chapter

Proposition 2. *Definition 3 and Definition 4 are equivalent.*

Proof. • It is obvious that Definition 4 implies Definition 3. We hence have to show that Definition 3 implies Definition 4:

- We consider G' the generating set of the group G, that is all element of G can be written as a finite combination (under the law of the group $*$) of elements of G' and their inverse.
- We now consider Definition 3 on the elements of G' only. Then we can build a group homomorphism from Definition 3 from the bijective transformations arising from G'. Indeed, consider $g \in G$, then $g = g_1 * g_2 * \ldots * g_k$ for $g_i \in G'$. We consider $T_g = T_{g_1} \circ T_{g_2} \circ \ldots \circ T_{g_k}$. It is clear that $T_{g^{-1}} = T_{g_k^{-1}} \circ T_{g_{k-1}^{-1}} \circ \ldots \circ T_{g_1^{-1}}$. Assume for the sake of simplicity $k = 2$. T_{g_2} is a bijective state-space transformation for the commutative relation to hold between f and f_{g_2}. T_{g_1} is a bijective state-space transformation for the commutative relation to hold between f_{g_2} and $f_{g_1 * g_2}$:

$$T_{g_2}^{-1} \mathcal{A}^{f_{g_2}} T_{g_2} = \mathcal{A}^{f} \tag{11.1}$$

$$T_{g_1}^{-1} \mathcal{A}^{f_{g_1 * g_2}} T_{g_1} = \mathcal{A}^{f_{g_2}} \tag{11.2}$$

  Hence $T_g = T_{g_1} \circ T_{g_2}$ satisfies the commutative relation from f to $f_{g_1 * g_2}$. Note that we have used that we have a group action and that $g_1.(g_2.f) = (g_1 * g_2).f$. This way we can build a homomorphism by defining the bijective transformation for elements of g that are the product of elements of G' as the composition of the elementary bijective transformations. Since the relation

$$T_g = T_{g_1} \circ T_{g_2}$$

  if $g = g_1 * g_2$ corresponds to the homomorphism property, we have build an homomorphism and proven that Definition 2 is satisfied.

□

11.2 Proof of Theorem 4

Proof. We need to take the limit for n to infinity of the following expectation

$$nF^{(n)}\left(\frac{\sigma}{n}\right) = \frac{1}{2} E\left[n \ln^{-}\left(1 - 2\frac{\sigma}{n}[\mathcal{N}]_1 + \frac{\sigma^2}{n^2} \underbrace{\|\mathcal{N}\|^2}_{\chi^2 \text{ dist.}} \right) \right]$$

where $\mathcal{N}$ follows a n-dimensional multivariate normal distribution. Let us define

$$Y_n = \frac{n}{2}\ln^-\left(1 - 2\frac{\sigma}{n}[\mathcal{N}]_1 + \frac{\sigma^2}{n^2}\sum_{i=1}^{n}[\mathcal{N}]_i^2\right)$$

such that $nF^{(n)}\left(\frac{\sigma}{n}\right) = E[Y_n]$. We have the following almost sure limit

$$\lim_{n\to\infty} Y_n = \left(\sigma[\mathcal{N}]_1 - \frac{\sigma^2}{2}\right)1_{\{-2[\mathcal{N}]_1+\sigma\leq 0\}} \ ,$$

where we use the fact that by the Strong Law of Large numbers $\frac{1}{n}\sum_{i=1}^{n}[\mathcal{N}]_i^2 = 1$. We now need to prove the uniform integrability of the family $(Y_n)_{n\geq 1}$. We use the fact that for $x > -1$

$$\begin{aligned} n\ln^-(1+x) = n\ln^-(1+x)1_{\{-1<x\leq 0\}} &= -n\ln(1+x)1_{\{-1<x\leq 0\}} = \ln\left[\left(\frac{1}{1+x}\right)^n\right]1_{\{-1<x\leq 0\}} \\ &= 8\ln\left[\left(\frac{1}{1+x}\right)^{n/8}\right]1_{\{-1<x\leq 0\}} \leq 8\left(\frac{1}{1+x}\right)^{n/8}1_{\{-1<x\leq 0\}} \end{aligned} \quad (11.3)$$

We apply the obtained inequality to Y_n and thus obtain the following bound:

$$Y_n = \frac{n}{2}\ln^-\left(1 - 2\frac{\sigma}{n}[\mathcal{N}]_1 + \frac{\sigma^2}{n^2}\|\mathcal{N}\|^2\right) \leq 4\left(\frac{1}{1 - 2\frac{\sigma}{n}[\mathcal{N}]_1 + \frac{\sigma^2}{n^2}\|\mathcal{N}\|^2}\right)^{n/8}1_{\{-1<-2\frac{\sigma}{n}[\mathcal{N}]_1+\frac{\sigma^2}{n^2}\|\mathcal{N}\|^2\leq 0\}}$$

To prove the uniform integrability of $(Y_n)_n$ we prove the that there exists C such that $E[|Y_n|^2] < C$ for all n. We have according to the previous inequality

$$E\left[|Y_n|^2\right] \leq 16E\left[\left(\frac{1}{1 - 2\frac{\sigma}{n}[\mathcal{N}]_1 + \frac{\sigma^2}{n^2}\|\mathcal{N}\|^2}\right)^{n/4}1_{\{-1<-2\frac{\sigma}{n}[\mathcal{N}]_1+\frac{\sigma^2}{n^2}\|\mathcal{N}\|^2\leq 0\}}\right]$$

We now apply spherical coordinates (assuming $n \geq 2$) to the RHS of the previous equation

$$\begin{aligned} E\left[|Y_n|^2\right] \leq \frac{16}{4W_{n-2}}\int_0^{\frac{\pi}{2}}\int_0^{+\infty}\left(\frac{1}{1 - 2\frac{\sigma}{n}\sqrt{r}\cos\theta + \frac{\sigma^2}{n^2}r}\right)^{n/4} & 1_{\{-1<-2\frac{\sigma}{n}\sqrt{r}\cos\theta+\frac{\sigma^2}{n^2}r\leq 0\}} \\ & \sin^{n-2}(\theta)\frac{\exp(-\frac{r}{2})r^{\frac{n}{2}-1}}{\Gamma(\frac{n}{2})2^{n/2}}d\theta dr \end{aligned} \quad (11.4)$$

From simple geometry we know that $1 - 2\frac{\sigma}{n}\sqrt{r}\cos\theta + \frac{\sigma^2}{n^2}r \geq \sin^2\theta$ such that

$$E\left[|Y_n|^2\right] \leq \frac{16}{4W_{n-2}}\int_0^{\frac{\pi}{2}}\int_0^{+\infty}\left(\frac{1}{\sin\theta}\right)^{n/2}\sin^{n-2}(\theta)\frac{\exp(-\frac{r}{2})r^{\frac{n}{2}-1}}{\Gamma(\frac{n}{2})2^{n/2}}d\theta dr \quad (11.5)$$

$$= \frac{16}{4W_{n-2}}\int_0^{\pi/2}\sin^{\frac{n}{2}-2}\theta d\theta = \frac{4W_{\frac{n}{2}-2}}{W_{n-2}} \leq 4(\sqrt{2}+1) \quad (11.6)$$

where the latter inequality holds for n large enough. Hence for all n, there exists C such that $E[|Y_n|^2] < \infty$ such that $(Y_n)_n$ is uniformly integrable. Hence we can conclude that

$$\begin{aligned} \lim_{n\to\infty} nF^{(n)}(\sigma/n) = E\left[\left(\sigma[\mathcal{N}]_1 - \frac{\sigma^2}{2}\right)1_{\{-2[\mathcal{N}]_1+\sigma\leq 0\}}\right] &= \sigma E[[\mathcal{N}]_1 1_{\{[\mathcal{N}]_1\geq\frac{\sigma}{2}\}}] - \frac{\sigma^2}{2}\Pr([\mathcal{N}]_1 \geq \frac{\sigma}{2}) \\ &= \frac{\sigma}{\sqrt{2\pi}}\exp\left(-\frac{\sigma^2}{8}\right) - \frac{\sigma^2}{2}\Phi\left(-\frac{\sigma}{2}\right) \end{aligned} \quad (11.7)$$

where for the latter inequality we have use Lemma 14 in [12]. □

Chapter 12

Notations - Abbreviations - Terminology

$[\mathbf{x}]_i$	i^{th} coordinate of a vector $\mathbf{x}$
e_1	first unit vector in $\mathbb{R}^n$, i.e. $e_1 = (1, 0, \ldots, 0)$
$\mathbb{N}$	natural numbers counting zero $\{0, 1, \ldots\}$
$\mathbb{N}_>$	natural numbers excluding zero $\{1, 2, \ldots\}$
$\mathbb{R}$	real numbers
$\mathbb{R}^+$	$\mathbb{R}^+ = [0, +\infty($
$\mathbb{R}^+_>$	$)0, +\infty($
$\mathcal{N}(0, 1)$	standard normal distribution
I_d	Identity matrix (in dimension n)
$\mathcal{N}(0, I_d)$	multivariate normal distribution with mean zero and covariance matrix identity
$\mathcal{N}(\mathbf{m}, \mathbf{C})$	multivariate normal distribution with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{C}$
$\mathrm{GL(n, \mathbb{R})}$	real $n \times n$ invertible matrices
$\mathrm{S}(n, \mathbb{R})$	real $n \times n$ symmetric positive definite matrix
$\mathrm{SO}(n, \mathbb{R})$	group special orthogonal
$\mathbf{C}^{1/2}$	square root of $\mathbf{C}$ that satisfies $\mathbf{C}^{1/2}[\mathbf{C}^{1/2}]^T = \mathbf{C}$ and is symmetric
$\mathcal{M}_I$	set of strictly monotone functions from $I \subset \mathbb{R} \to \mathbb{R}$
$\mathcal{M} = \bigcup_{I \subset \mathbb{R}} \mathcal{M}_I$	
$A \subset B$	A is a subset of B that can also be equal to B
sphere function	$\mathbf{x} \in \mathbb{R}^n \mapsto \sum_{i=1}^n \mathbf{x}_i^2 = \|\mathbf{x}\|^2$
spherical function	$\mathbf{x} \in \mathbb{R}^n \mapsto g(\sum_{i=1}^n \mathbf{x}_i^2)$ where $g \in \mathcal{M}_{\mathbb{R}^+}$
i.i.d.	independent identically distributed
w.l.g.	without loss of generality
w.r.t.	with respect to
w.l.o.g.	without loss of generality
RHS	right hand side
LHS	left hand side
DFO	Derivative-Free Optimization
EC	Evolutionary Computation
EA	Evolutionary Algorithms
ES	Evolution Strategies
CMA	Covariance Matrix Adaptation
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
CSA	Cumulative Step-size Adaptation (default step-size mechanism of the CMA-ES algorithm)
IGO	Information Geometric Optimization

GA	Genetic Algorithm
NEWUOA	NEW Unconstraint Optimization Algorithm
CB-SARS	Comparison-Based Step-Size Adaptive Randomized Search

Chapter 13

Bibliography

[1] Ouassim Ait Elhara, Anne Auger, and Nikolaus Hansen. A Median Success Rule for Non-Elitist Evolution Strategies: Study of Feasibility. In Blum et al. Christian, editor, *Genetic and Evolutionary Computation Conference*, pages 415–422, Amsterdam, Netherlands, July 2013. ACM, ACM Press.

[2] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi. Bidirectional relation between CMA evolution strategies and natural evolution strategies. In R. Schaefer et al., editors, *Parallel Problem Solving from Nature (PPSN XI)*, volume 6238 of *Lecture Notes in Computer Science*, pages 154–163. Springer Verlag, 2010.

[3] Youhei Akimoto, Anne Auger, and Nikolaus Hansen. Convergence of the Continuous Time Trajectories of Isotropic Evolution Strategies on Monotonic C^2-composite Functions. In *PPSN - 12th International Conference on Parallel Problem Solving from Nature - 2012*, Taormina, Italy, September 2012. Springer.

[4] Youhei Akimoto, Anne Auger, and Nikolaus Hansen. Comparison-Based Natural Gradient Optimization in High Dimension. In *Genetic and Evolutionary Computation Conference GECCO'14*, Vancouver, Canada, July 2014. ACM.

[5] Dirk V. Arnold. Optimal weighted recombination. In *Foundations of Genetic Algorithms 8*, pages 215–237. Springer Verlag, 2005.

[6] Charles Audet and John E Dennis Jr. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903, 2002.

[7] A. Auger. Convergence results for (1,λ)-SA-ES using the theory of φ-irreducible markov chains. *Theoretical Computer Science*, 334(1-3):35–69, 2005.

[8] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Articulating User Preferences in Many-Objective Problems by Sampling the Weighted Hypervolume. In G. Raidl et al., editors, *Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 555–562, New York, NY, USA, 2009. ACM.

[9] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Investigating and Exploiting the Bias of the Weighted Hypervolume to Articulate User Preferences. In G. Raidl et al., editors, *Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 563–570, New York, NY, USA, 2009. ACM.

[10] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the Hypervolume Indicator: Optimal μ-Distributions and the Choice of the Reference Point. In *Foundations of Genetic Algorithms (FOGA 2009)*, pages 87–102, New York, NY, USA, 2009. ACM.

[11] A. Auger, D. Brockhoff, and N. Hansen. Benchmarking the (1,4)-CMA-ES With Mirrored Sampling and Sequential Selection on the Noisy BBOB-2010 Testbed. In *GECCO (Companion) workshop on Black-Box Optimization Benchmarking (BBOB'2010)*, pages 1625–1632. ACM, 2010.

[12] A. Auger, D. Brockhoff, and N. Hansen. Analyzing the impact of mirrored sampling and sequential selection in elitist evolution strategies. In *Foundations of Genetic Algorithms (FOGA 11)*, pages 127–138. ACM Press, 2011.

[13] A. Auger, D. Brockhoff, and N. Hansen. Mirrored sampling in evolution strategies with weighted recombination. In *Genetic and Evolutionary Computation Conference (GECCO 2011)*, pages 861–868. ACM Press, 2011.

[14] A. Auger, D. Brockhoff, and N. Hansen. Benchmarking the Local Metamodel CMA-ES on the Noiseless BBOB?2013 Test Bed. In *GECCO (Companion) workshop on Black-Box Optimization Benchmarking (BBOB'2013)*, pages 1225–1232. ACM, 2013.

[15] A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pages 1777–1784 Vol. 2, Sept 2005.

[16] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 1769–1776. IEEE Press, 2005.

[17] A. Auger and N. Hansen. Reconsidering the progress rate theory for evolution strategies in finite dimensions. In *Genetic and Evolutionary Computation Conference (GECCO 2006)*, pages 445–452. ACM Press, 2006.

[18] A. Auger and N. Hansen. Linear convergence on positively homogeneous functions of a comparison based step-size adaptive randomized search: the (1+1) ES with generalized one-fifth success rule, 2013. ArXiv eprint.

[19] A. Auger, N. Hansen, J. Perez Zerpa, R. Ros, and M. Schoenauer. Experimental comparisons of derivative free optimization algorithms. In Jan Vahrenhold, editor, *8th International Symposium on Experimental Algorithms*, volume 5526 of *LNCS*, pages 3–15. Springer, 2009.

[20] A. Auger, N. Hansen, J. M. Perez Zerpa, R. Ros, and M. Schoenauer. Empirical comparisons of several derivative free optimization algorithms. In *Acte du 9ime colloque national en calcul des structures*, volume 1, pages 481–486, 2009.

[21] Anne Auger, Johannes Bader, and Dimo Brockhoff. Theoretically Investigating Optimal μ-Distributions for the Hypervolume Indicator: First Results For Three Objectives. In *Parallel Problem Solving from Nature (PPSN XI)*, Krakow, Poland, September 2010.

[22] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*, 425(0):75 – 103, 2012. Theoretical Foundations of Evolutionary Computation.

[23] Anne Auger, Philippe Chatelain, and Petros Koumoutsakos. R-leaping: Accelerating the stochastic simulation algorithm by reaction leaps. *J. Chem. Phys.*, 125(8):084103+, 2006.

[24] Anne Auger and Nikolaus Hansen. On Proving Linear Convergence of Comparison-based Step-size Adaptive Randomized Search on Scaling-Invariant Functions via Stability of Markov Chains, 2013. ArXiv eprint.

[25] S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithms. In A. Prieditis and S. Russel, editors, *ICML95*, pages 38–46. Morgan Kaufmann, 1995.

[26] H.-G. Beyer. *The Theory of Evolution Strategies*. Springer Verlag, 2001.

[27] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies – a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[28] Vivek S Borkar. Stochastic approximation: a dynamical systems viewpoint. Cambridge University Press, 2008.

[29] Zyed Bouzarkouna, Anne Auger, and Didier Yu Ding. Investigating the Local-Meta-Model CMA-ES for Large Population Sizes. In Cecilia Di Chio, Stefano Cagnoni, Carlos Cotta, Marc Ebner, Anikó Ekárt, Anna Esparcia-Alcázar, Chi Keong Goh, and Juan J, editors, *3rd European event on Bio-inspired algorithms for continuous parameter optimisation (EvoNUM'10)*, volume 6024 of *Lecture Notes in Computer Science*, pages 402–411, Istanbul, Turkey, April 2010.

[30] Zyed Bouzarkouna, Anne Auger, and Didier Yu Ding. Local-Meta-Model CMA-ES for Partially Separable Functions. In *Genetic and Evolutionary Computation Conference (GECCO 2011)*, pages 869–876, Dublin, Ireland, July 2011.

[31] Zyed Bouzarkouna, Didier Yu Ding, and Anne Auger. Partially Separated Meta-models with Evolution Strategies for Well Placement Optimization. In *73rd EAGE Conference & Exhibition incorporating SPE EUROPEC*, pages 1–9, Vienna, Austria, May 2011.

[32] Zyed Bouzarkouna, Didier Yu Ding, and Anne Auger. Well Placement Optimization with the Covariance Matrix Adaptation Evolution Strategy and Meta-Models. *Computational Geosciences*, 16(1):75–92, September 2011.

[33] D. Brockhoff, A. Auger, and N. Hansen. On the Impact of a Small Initial Population Size in the IPOP Active CMA-ES with Mirrored Mutations on the Noiseless BBOB Testbed. In *GECCO (Companion) workshop on Black-Box Optimization Benchmarking (BBOB'2012)*. ACM, 2012. accepted for publication.

[34] Dimo Brockhoff, Anne Auger, Nikolaus Hansen, Dirk V. Arnold, and Tim Hohm. Mirrored Sampling and Sequential Selection for Evolution Strategies. In *PPSN*, Parallel Problem Solving from Nature (PPSN XI), pages 11–21, Warsaw, Poland, September 2010.

[35] A. Chotard, A. Auger, and N. Hansen. Cumulative step-size adaptation on linear functions. In *Parallel Problem Solving from Nature - PPSN XII*, pages 72–81. Springer, 2012.

[36] Alexandre Chotard, Anne Auger, and Nikolaus Hansen. Cumulative Step-size Adaptation on Linear Functions: Technical Report. Research report, June 2012.

[37] Alexandre Chotard, Anne Auger, and Nikolaus Hansen. Markov chain analysis of cumulative step-size adaptation on a linear constraint problem. *Evolutionary Computation*, 2015. accepted.

[38] A.A.R. Conn, K. Scheinberg, and L.N. Vicente. *Introduction to Derivative-free Optimization*. MPS-SIAM series on optimization. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2009.

[39] K. Deb, M. Mohan, and S. Mishra. Evaluating the ϵ-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation*, 13(4):501–525, Winter 2005.

[40] Peter Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, volume 35. Springer, 2011.

[41] L. Devroye. The compound random search. In *International Symposium on Systems Engineering and Analysis*, pages 195–110. Purdue University, 1972.

[42] M. Emmerich, N. Beume, and B. Naujoks. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 3410 of *LNCS*, pages 62–76. Springer, 2005.

[43] T. Glasmachers, T. Schaul, Y. Sun, D. Wierstra, and J. Schmidhuber. Exponential natural evolution strategies. In *Genetic and Evolutionary Computation Conference (GECCO 2010)*, pages 393–400. ACM Press, 2010.

[44] N. Hansen. An analysis of mutative σ-self-adaptation on linear fitness functions. *Evolutionary Computation*, 14(3):255–275, 2006.

[45] N. Hansen, D. V. Arnold, and A. Auger. Evolution strategies. In Janusz Kacprzyk and Witold Pedrycz, editors, *Handbook of Computational Intelligence*, chapter 44. Springer, 2015.

[46] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Technical Report RR-6828, INRIA, 2009.

[47] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009. Updated February 2010.

[48] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noisy functions definitions. Technical Report RR-6869, INRIA, 2009. Updated February 2010.

[49] N. Hansen, F. Gemperle, A. Auger, and P. Koumoutsakos. When do heavy-tail distributions help? In T. P. Runarsson et al., editors, *Parallel Problem Solving from Nature PPSN IX*, volume 4193 of *Lecture Notes in Computer Science*, pages 62–71. Springer, 2006.

[50] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[51] Nikolaus Hansen, Asma Atamna, and Anne Auger. How to Assess Step-Size Adaptation Mechanisms in Randomised Search. In T. Bartz-Beielstein et al, editor, *Parallel Problem Solving from Nature, PPSN XIII*, volume 8672 of *LNCS*, pages 60–69, Ljubljana, Slovenia, September 2014. Springer.

[52] Nikolaus Hansen and Anne Auger. Principled design of continuous stochastic search: From theory to practice. In Springer, editor, *Theory and Principled Methods for Designing Metaheuristics*, pages 145–180. Y. Borenstein and A. Moraglio, 2014.

[53] Nikolaus Hansen, Raymond Ros, Nikolas Mauny, Marc Schoenauer, and Anne Auger. Impacts of Invariance in Search: When CMA-ES and PSO Face Ill-Conditioned and Non-Separable Problems. *Applied Soft Computing*, 11:5755–5769, 2011.

[54] R. Hooke and T.A. Jeeves. "Direct Search" Solution of Numerical and Statistical Problems. *Journal of the ACM*, 8:212–229, 1961.

[55] J.N. Hooker. Testing heuristics: We have it all wrong. *Journal of Heuristics*, 1:33–42, 1995.

[56] Jens Jägersküpper. Rigorous runtime analysis of the (1+1)-ES: 1/5-rule and ellipsoidal fitness landscapes. In LNCS, editor, *Foundations of Genetic Algorithms: 8th International Workshop, FoGA 2005*, volume 3469, pages 260–281, 2005.

[57] Jens Jägersküpper. How the (1+1) ES using isotropic mutations minimizes positive definite quadratic forms. *Theoretical Computer Science*, 361(1):38–56, 2006.

[58] Jens Jägersküpper. Probabilistic runtime analysis of $(1\overset{+}{,}\lambda)$ evolution strategies using isotropic mutations. In *Genetic and Evolutionary Computation Conference (GECCO 2006)*, pages 461–468. ACM Press, 2006.

[59] Jens Jägerskϋpper. Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. *Theoretical Computer Science*, 379(3):329–347, 2007.

[60] M. Jebalia, A. Auger, and N. Hansen. Log-linear convergence and divergence of the scale-invariant (1+1)-ES in noisy environments. *Algorithmica*, 59(3):425–460, 2011.

[61] M. Jebalia, A. Auger, and P. Liardet. Log-linear convergence and optimal bounds for the (1+1)-ES. In N. Monmarché et al., editors, *Evolution Artificielle (EA '07)*, volume 4926 of *LNCS*, pages 207–218. Springer Verlag, 2008.

[62] Mohamed Jebalia and Anne Auger. Log-linear Convergence of the Scale-invariant $(\mu/\mu\_w, \lambda)$-ES and Optimal μ for Intermediate Recombination for Large Population Sizes. In Robert Schaefer, Carlos Cotta, Joanna Kolodziej, and Günter Rudolph, editors, *Parallel Problem Solving From Nature (PPSN2010)*, Lecture Notes in Computer Science, pages 52–61, Krakow, Poland, September 2010. Springer.

[63] Mohamed Jebalia and Anne Auger. Log-linear Convergence of the Scale-invariant $(\mu/\mu\_w, \lambda)$-ES and Optimal mu for Intermediate Recombination for Large Population Sizes. Research Report RR-7275, June 2010.

[64] C. Kappler. Are evolutionary algorithms improved by large mutations? In H.-M. Voigt et al., editors, *Parallel Problem Solving from Nature (PPSN IV)*, volume 1141 of *Lecture Notes in Computer Science*, pages 346–355. Springer Verlag, 1996.

[65] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948, 1995.

[66] S. Kern, N. Hansen, and P. Koumoutsakos. Local meta-models for optimization using evolution strategies. In T. Runarsson et al., editor, *Parallel Problem Solving from Nature - PPSN IX*, volume 4193 of *Lecture Notes in Computer Science*, pages 939–948. Springer Verlag, 2006.

[67] J. Knowles and D. Corne. Properties of an Adaptive Archiving Algorithm for Storing Non-dominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, 2003.

[68] J. D. Knowles, D. W. Corne, and M. Fleischer. Bounded Archiving using the Lebesgue Measure. In *Congress on Evolutionary Computation (CEC 2003*, pages 2490–2497, Canberra, Australia, 2006. IEEE Press.

[69] Harold J. Kushner and G. George Yin. *Stochastic approximation and recursive algorithms and applications*. Springer Verlag, 2nd edition, 2003.

[70] Jeffrey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147, 1998.

[71] Pedro Larraanaga and Jose A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[72] Robert M Lewis and Virginia Torczon. Rank ordering and positive bases in pattern search algorithms. Technical report, DTIC Document, 1996.

[73] G. Lizarraga-Lizarraga, A. Hernandez-Aguirre, and S. Botello-Rionda. G-Metric: an M-ary quality indicator for the evaluation of non-dominated sets. In *Genetic And Evolutionary Computation Conference (GECCO 2008)*, pages 665–672, New York, NY, USA, 2008. ACM.

[74] Lennart Ljung. Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, 22(4):551–575, 1977.

[75] Ilya Loshchilov, Marc Schoenauer, and Michèle Sebag. Achieving optimization invariance w.r.t. monotonous transformations of the objective function and orthogonal transformations of the representation. In *Probabilistic Numerics Workshop of NIPS 2012*, 2012.

[76] Philippe Martin, Pierre Rouchon, and Joachim Rudolph. Invariant tracking. *ESAIM: Control, Optimisation and Calculus of Variations*, 10(01):1–13, 2004.

[77] Ken IM McKinnon. Convergence of the nelder–mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 9(1):148–158, 1998.

[78] S.P. Meyn and R.L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, New York, 1993.

[79] John Ashworth Nelder and R Mead. A simplex method for function minimization. *The Computer Journal*, pages 308–313, 1965.

[80] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *ArXiv e-prints*, June 2013.

[81] Michael J.D. Powell. The newuoa software for unconstrained optimization without derivatives. Technical Report DAMTP 2004/NA05, CMS, University of Cambridge, Cambridge CB3 0WA, UK, November 2004.

[82] Michael J.D. Powell. Developments of newuoa for unconstrained minimization without derivatives. Technical Report DAMTP 2007/NA05, CMS, University of Cambridge, Cambridge CB3 0WA, UK, June 2007.

[83] I. Rechenberg. *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart, 1973.

[84] I. Rechenberg. *Evolutionsstrategie '94*. Frommann-Holzboog Verlag, 1994.

[85] G. Rudolph. Local convergence rates of simple evolutionary algorithms with cauchy mutations. *IEEE Transactions on Evolutionary Computation*, 1(4):249–258, 1997.

[86] M. Schumer and K. Steiglitz. Adaptive step size random search. *Automatic Control, IEEE Transactions on*, 13:270–276, 1968.

[87] M. Schumer and K. Steiglitz. Adaptive step size random search. *IEEE Transactions on Automatic Control*, 13(3):270–276, 1968.

[88] J. C. Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *Automatic Control, IEEE Transactions on*, 45(10):1839–1853, 2000.

[89] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4):341–359, 1997.

[90] Virginia Torczon. On the convergence of pattern search algorithms. *SIAM Journal on optimization*, 7(1):1–25, 1997.

[91] X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102, 1999.

[92] G. George Yin, Günter Rudolph, and Hans-Paul Schwefel. Establishing connections between evolutionary algorithms and stochastic approximation. *Informatica*, 1:93–116, 1995.

[93] G. George Yin, Günter Rudolph, and Hans-Paul Schwefel. Analyzing the (1, λ) evolution strategy via stochastic approximation methods. *Evolutionary Computation*, 3(4):473–489, 1996.

[94] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In *Conference on Parallel Problem Solving from Nature (PPSN V)*, pages 292–301, Amsterdam, 1998.